

LEVEL III

12

AD A059538

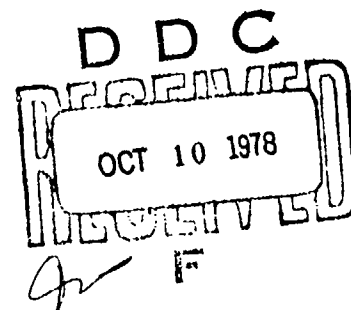
DDC FILE COPY

# STRUCTURAL AREA INSPECTION FREQUENCY EVALUATION (SAIFE)

Volume IV. Software Documentation and User's Manual

Book 1. Initial Program

Carter J. Dinkeloo  
Martin S. Moran



APRIL 1978  
FINAL REPORT

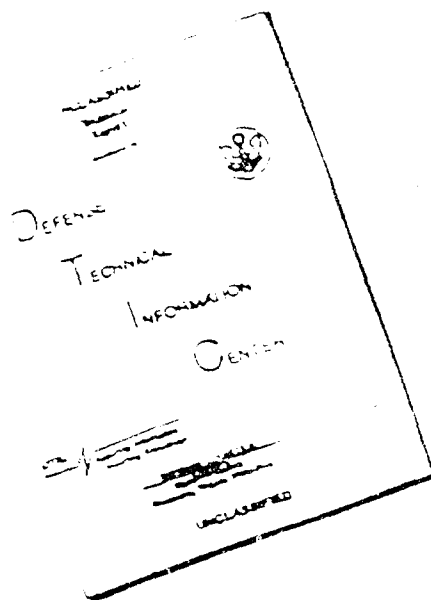
Document is available to the U.S. public through  
the National Technical Information Service,  
Springfield, Virginia 22161.

Prepared for

**U.S. DEPARTMENT OF TRANSPORTATION**  
**FEDERAL AVIATION ADMINISTRATION**  
**Systems Research & Development Service**  
**Washington, D.C. 20590**



# DISCLAIMER NOTICE



THIS DOCUMENT IS BEST  
QUALITY AVAILABLE. THE COPY  
FURNISHED TO DTIC CONTAINED  
A SIGNIFICANT NUMBER OF  
PAGES WHICH DO NOT  
REPRODUCE LEGIBLY.

REPRODUCED FROM  
BEST AVAILABLE COPY

NOTICE

This document is disseminated under the sponsorship of the Department of Transportation in the interest of information exchange. The United States Government assumes no liability for its contents or use thereof.

1. Report No. FAA-RD-78-29, IV, Book 1		2. Government Accession No. (SAIFE)		3. Recipient's Catalog No.	
4. Title and Subtitle STRUCTURAL AREA INSPECTION FREQUENCY EVALUATION Volume IV. Software Documentation and User's Manual. Book 1 - Initial Program,				5. Report Date April 1978	
7. Author(s) Carter J. Dinkeloo, Martin S. Moran				6. Performing Organization Code	
9. Performing Organization Name and Address Technology Incorporated P.O. Box 3036, Overlook Branch Dayton, Ohio 45431				8. Performing Organization Report No.	
12. Sponsoring Agency Name and Address Department of Transportation Federal Aviation Administration Systems Research and Development Service Washington, D.C. 20590				10. Work Unit No. (TRAIL)	
15. Supplementary Notes				11. Contract or Grant No. DOT-FA74WA-3493	
16. Abstract To assist in the evaluation of proposed structural inspection programs for commercial jet transport aircraft, a logic was developed to simulate structural defects, failures, and inspections. This logic was incorporated in a computer program entitled Structural Area Inspection Frequency Evaluation (SAIFE). With the objective of quantifying the evaluation process currently used to establish and modify inspection intervals, SAIFE accounts for the following factors: (1) aircraft design analysis; (2) fatigue testing; (3) production, service, and corrosion defects; (4) probability of crack or corrosion detection; and (5) aircraft modification economics. As a five-volume document, this report covers the initial contract effort plus a subsequent parametric analysis as follows: Volume I (entitled Executive Summary) presents the SAIFE logic and documents the methodology for the decision-making processes in the simulation logic. Volume II (entitled Description of Simulation Logic) details the SAIFE simulation logic, presents the background data for the analytical functions and decision-making processes, and includes data for a typical simulation problem. Volume III (entitled Demonstration Input, Inspection Survey, and MRR Data) presents data tabulations derived from historical trends and design input data for a SAIFE demonstration problem. As the user's manual for the SAIFE computer program, Volume IV contains detailed computer logic flow diagrams and a complete listing of the program which is written in SIMSCRIPT II.5. Volume V (entitled Results of Model Demonstration) presents the results of the program application to a hypothetical aircraft and compares these results with the service experience of operational aircraft.				13. Type of Report and Period Covered Final Report	
17. Key Words Aircraft experience simulation; aircraft inspection intervals; aircraft hazard rate and reliability; production, service, and corrosion defects; crack or corrosion detection probability; crack growth.				14. Sponsoring Agency Code	
18. Distribution Statement Document is available to the public through the National Technical Information Service, Springfield, Virginia 22151.					
19. Security Classif. (of this report) UNCLASSIFIED		20. Security Classif. (of this page) UNCLASSIFIED		21. No. of Pages 209	
				22. Price	

# METRIC CONVERSION FACTORS

## Approximate Conversions to Metric Measures

Symbol When You Know Multiply by To Find Symbol

### LENGTH

in Centimeters  
ft meters  
yd kilometers  
mi

### AREA

sq in square centimeters  
sq ft square meters  
sq yd square meters  
sq mi square kilometers  
acres hectares

### MASS (weight)

oz grams  
lb kilograms  
short tons (2000 lb) tonnes

### VOLUME

tsp milliliters  
 Tbsp milliliters  
 fl oz milliliters  
 c liters  
 pt liters  
 qt liters  
 gal liters  
 cu ft cubic meters  
 cu yd cubic meters

### TEMPERATURE (exact)

Fahrenheit temperature 5/9 (after subtracting 32) Celsius temperature °C

## Approximate Conversions from Metric Measures

Symbol When You Know Multiply by To Find Symbol

### LENGTH

mm inches  
cm inches  
m feet  
km miles

### AREA

square centimeters square inches  
square meters square meters  
square kilometers square meters  
hectares (10,000 m<sup>2</sup>) square meters

### MASS (weight)

g grams  
kg kilograms  
tonnes (1000 kg) short tons

### VOLUME

ml milliliters  
l liters  
cl centiliters  
cubic meters  
cubic centimeters

### TEMPERATURE (exact)

Celsius temperature 9/5 (then add 32) Fahrenheit temperature °F



\* 1 in = 2.54 (exactly). For other exact conversions and more data, visit [www.nist.gov](http://www.nist.gov), P. 20-196. Units of Weights and Measures, Price \$2.25, SO Catalog No. C1310 296.

## PREFACE

Technology Incorporated prepared this fourth volume of a five-volume report to document the simulation logic for the Structural Area Inspection Frequency Evaluation (SAIFE) in accordance with Article II, paragraph B of Contract DOT-FA74WA-3493. (Volume IV along with Volume V completes the requirements of Phase III of the contract.) The effort is sponsored by the Aircraft Safety and Noise Abatement Division, Systems Research and Development Service of the Federal Aviation Administration.

The principal Technology Incorporated personnel engaged on this program were Mr. Carter J. Dinkeloo, project engineer, who served as principal investigator; Mr. Martin S. Moran, research engineer, who developed the model for the SAIFE computer program; and Mr. Ronald I. Rockafellow, program manager.

The contract monitors for the FAA were Messrs. Herbert Spicer and Charles Troha of the Aircraft Safety and Noise Abatement Division. The technical monitor was Mr. Arnold E. Anderjaska of the Flight Standards Division.

**A 23**

## TABLE OF CONTENTS

<u>Section</u>	<u>Page</u>
I            INTRODUCTION . . . . .	1
II           SYSTEM DESCRIPTION . . . . .	3
III          GENERAL PROGRAM DESCRIPTION . . . . .	5
IV          OPERATING PROCEDURES . . . . .	6
1.           Computer Requirements . . . . .	6
2.           Input . . . . .	6
2.1           Aircraft Data . . . . .	7
2.2           Long List Data . . . . .	9
2.3           Element Data . . . . .	10
2.4           Format Specifications . . . . .	12
3.           Output . . . . .	13
3.1           Element Data . . . . .	14
3.2           Summary Data . . . . .	22
3.3           Long List Data . . . . .	22
APPENDIX A. DETAILED PROGRAM DESCRIPTION . . . . .	31
APPENDIX B. PROGRAM SOURCE LISTING . . . . .	119



## LIST OF ILLUSTRATIONS

<u>Figure</u>		<u>Page</u>
1	Approach to SAIPE Simulation . . . . .	2
2	Major Aspects of SAIPE Logic . . . . .	4
3	Sample Input Data to Produce Standard Output . .	13
4	Sample Input Data to Produce Long List Output .	14
5	Sample Standard Output Produced by Input Shown in Figure 3 . . . . .	15
6	Sample Long List Output Produced by Input Shown in Figure 4 . . . . .	24

## LIST OF TABLES

<u>Table</u>		<u>Page</u>
1	Initial Random Roots . . . . .	7

## I. INTRODUCTION

It is the mutual goal of the FAA, airframe manufacturers, and air carriers to constantly improve the structural integrity and inspection efficiency of civil aircraft. The good safety record of U.S. air carriers indicates that the current process of establishing and modifying structural inspection programs has been successful. However, with the increasing size and complexity of second- and third-generation transport aircraft, there is a need to quantify more precisely the present, subjective evaluation process which relies heavily on reliability analyses of the new design and on operational experience of similar aircraft.

Because of the extreme complexity of the evaluation process, a computer simulation of all critical aircraft service life aspects was judged the most rational means for quantifying the process more exactly. As a five-volume document, this report documents the resultant Structural Area Inspection Frequency Evaluation (SAIFE) simulation logic. SAIFE accounts for the following factors: (1) aircraft design analysis; (2) component and full-scale fatigue testing; (3) production, service, and corrosion defects; (4) probability of crack or corrosion detection; and (5) aircraft modification economics. It treats these factors in a logical sequence that realistically represents the procedure currently used to establish and modify inspection intervals. Figure 1 illustrates the data sources and analytical functions that are integrated into the SAIFE logic. SAIFE is designed to provide a repeatable method for evaluating proposed inspection programs. However, it is not intended to supplant the Maintenance Review Board or the air carrier use of the Standard Operations Specification - Aircraft Maintenance.

As Volume IV, this user's manual for the SAIFE program contains a system description, a program description, operating procedures, a sample input and output, and a source listing of the program. The detailed description of the program events and routines is presented in Appendix A, and the program source listing is contained in Appendix B which because of its voluminous computer-generated data is on the included microfiche.

The original computer model developed during the initial contract has been modified by the Engineering and Manufacturing Branch, Flight Standards National Field Office for the parametric study and is documented in Book 2 of this volume.

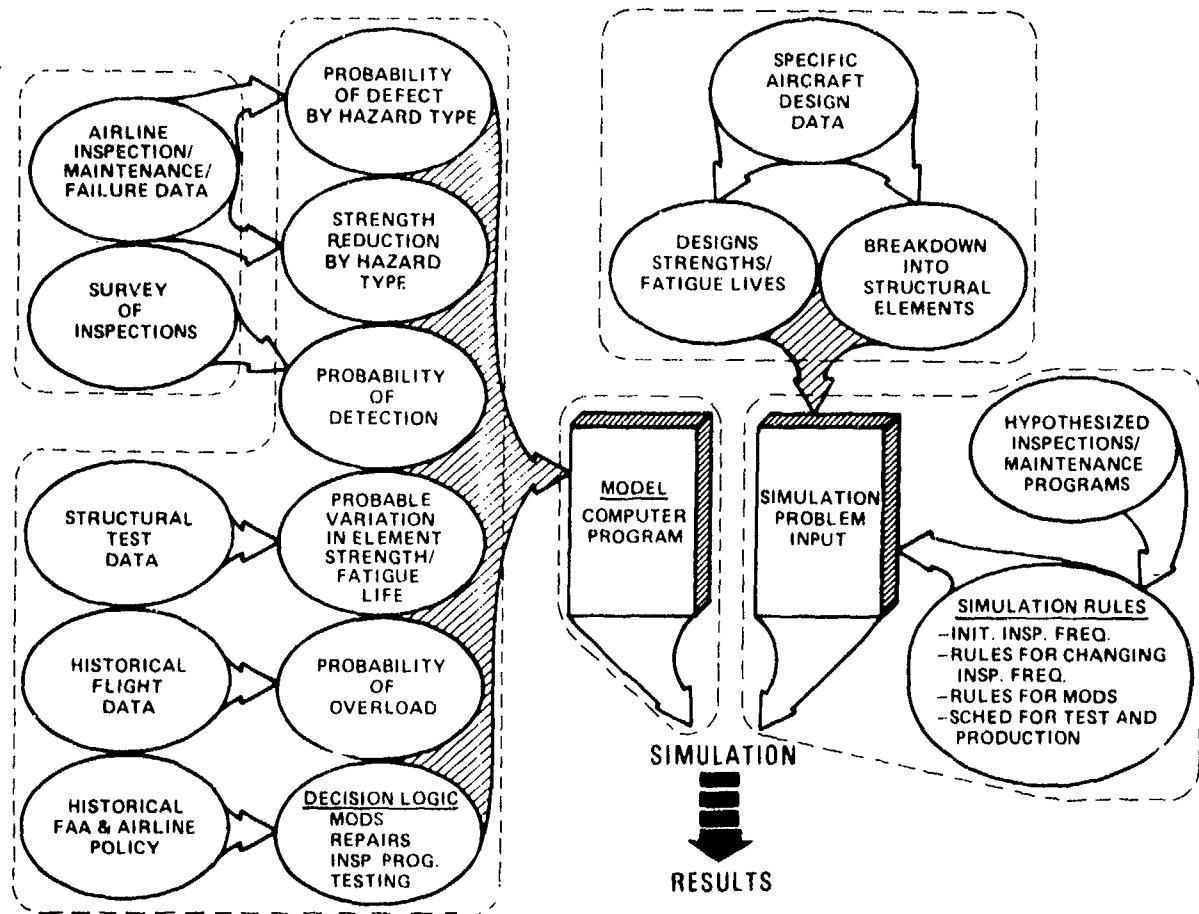


Figure 1. Approach to SAIPE Simulation

## II. SYSTEM DESCRIPTION

The eight blocks in Figure 2 represent the major aspects of the SAIFE simulation logic. Block 1 accepts input data for the aircraft fleet and for each structural element in the aircraft. After determining whether element modifications are required because of the fatigue test results in Block 2, Block 1 assigns a fatigue life to each element in each aircraft. Block 3 determines whether production, service, or corrosion defects will occur; if it is determined that such defects will occur, Block 3 predicts the times when they will occur. After comparing the flight loads with the strength of each element, Block 4 predicts the time to failure for each element. Block 5 conducts the periodic inspections of each element. If defects are detected, Block 6 repairs the element and assigns it a new fatigue life. However, if an existing defect is allowed to grow until element failure, Block 5 deletes the aircraft from the fleet. Depending on the magnitude of the detected defects, special inspections and increased inspection frequencies may be called for in Block 7 and modifications may be instituted in Block 8. When all the aircraft have been deleted from the fleet or retired from service, the simulation is complete.

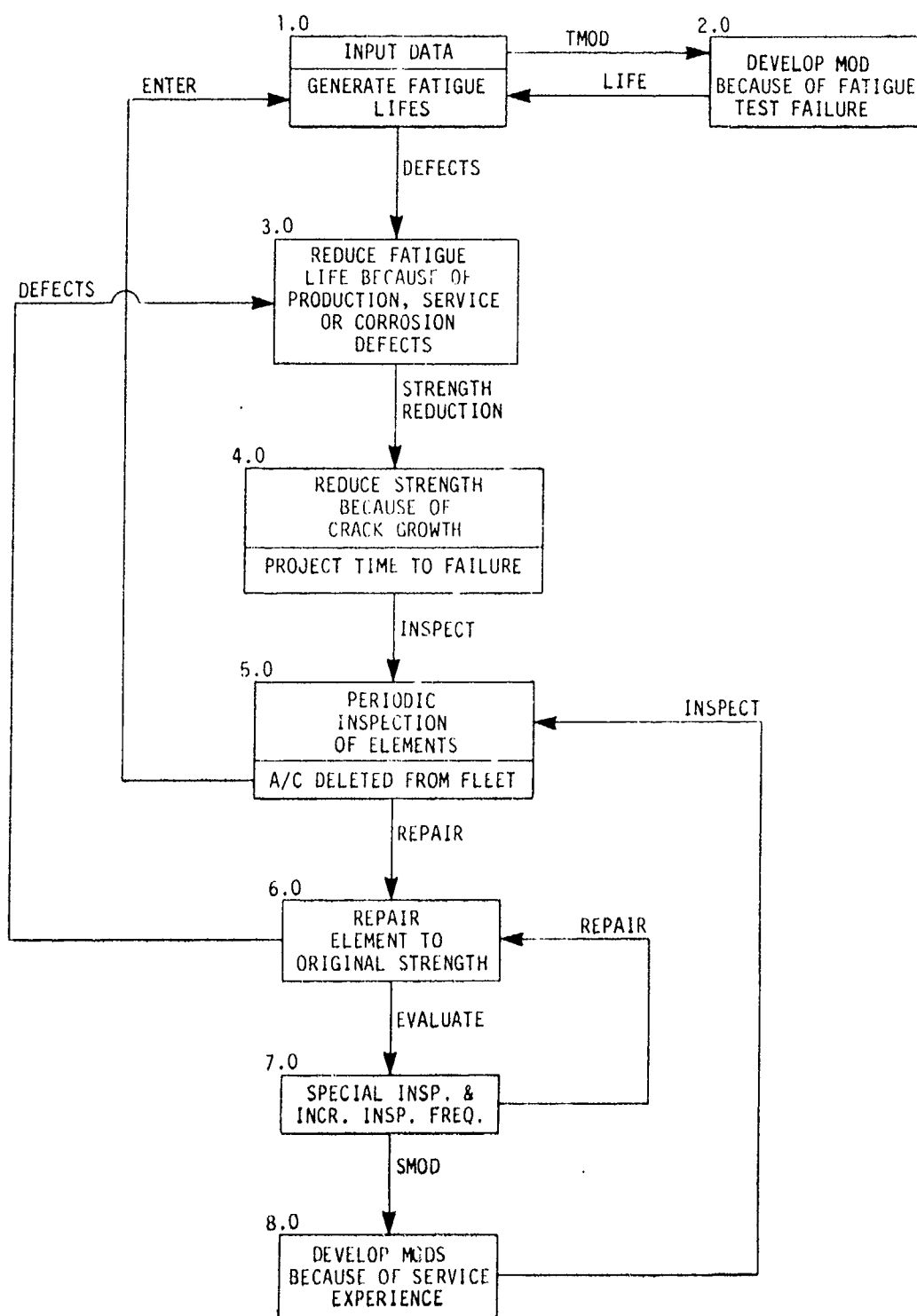


Figure 2. Major Aspects of SAIPE Logic

### III. GENERAL PROGRAM DESCRIPTION

The SAIFE program is a large, complex math model designed to simulate the structural performance of aircraft in a fleet of aircraft and the effectiveness of the inspection program for the aircraft fleet. The aircraft model included in the math model is divided into structurally significant elements and the inspection program for each element is defined. Structural defects are classified as follows: fatigue and corrosion which are wear-out and aging phenomena; production or design defects; and operational or maintenance damage. These defects and the inspection program are treated as probabilistic phenomena interacting over time. If the simulation is to proceed properly, the passage of simulated time must be controlled. This control can be accomplished by a user-designed algorithm or it can be done automatically with a "simulation clock" in one of the special purpose simulation languages. SIMSCRIPT II.5 <sup>T</sup> (trademark, Consolidated Analysis Centers Inc., Los Angeles, California), a computer language designed for discrete-event simulation applications, was chosen for the SAIFE simulation. SIMSCRIPT II.5 is a large language designed to facilitate the simulation of large, complex systems, and to reduce the total time spent in designing, programming, and testing simulation models.

Because of the extensive detail, the program events and routines are described in Appendix A.

#### IV. OPERATING PROCEDURES

##### 1. Computer Requirements

The SAIFE demonstration output presented in Volume V of this report was run on an IBM Model 360-65 computer using the SIMSCRIPT II.5 compiler, release 8F. The execution time per element ranged between 12 and 20 cpu seconds, and the compilation time was 205 cpu seconds. None of the demonstration runs required more than 325K bytes of core storage.

The run time and the core storage requirements depend on the values of the input variables. The variable that probably has the greatest effect on these requirements is the actual average fatigue life. The shorter the life, the greater the requirement: A shorter actual average fatigue life increases crack initiations across the fleet. These additional crack initiations, in turn, increase the inspection process, the repair activities, and the modification and interval change decisions. All these additional events require more execution time and storage space for event notices. The actual average fatigue life is shortened when the parameters involved in the distribution of the ratio of the actual to the predicted average fatigue life are so changed that the ratio is decreased.

Other variables which affect the run time and core storage requirements are as follows: with the actual average fatigue life established for all aircraft in the fleet, the requirements are increased if the parameters involved in the Weibull distribution of actual fatigue lives for individual aircraft are so changed that some lives are shortened and more crack initiations are introduced. In addition, regardless of the fatigue lives, the run time and storage requirements are increased if the defect occurrence rates are increased or the initial inspection intervals are decreased.

Since the seeds of the random number streams are initialized to the same values at the start of each execution, the user can reproduce the output, if desired, and more easily identify effects because of changes in the input parameters. Table 1 lists the ten seeds used by the IBM 360-65 to generate the demonstration output presented in Volume V.

##### 2. Input

The program input consists of three parts. The first part contains input variables which pertain to the aircraft type under consideration. These variables are input only once per simulation run and are constant from element to element. The second part is optional and is explained in Section IV,2.2. The third part contains input variables whose values are unique to each element. These variables must be input in their entirety for each element being simulated.

TABLE 1. INITIAL RANDOM ROOTS

SEED.V(1) = 2116429302  
SEED.V(2) = 683743814  
SEED.V(3) = 964393174  
SEED.V(4) = 1217426631  
SEED.V(5) = 618433579  
SEED.V(6) = 1157240309  
SEED.V(7) = 15726055  
SEED.V(8) = 48108509  
SEED.V(9) = 1797920909  
SEED.V(10) = 477424540

## 2.1 Aircraft Data

The input variables which pertain to the aircraft type are listed and described below in the order in which they are read in by SAIFE.

MODEL(\*) - This one-dimensional alpha array of size two identifies the aircraft type under consideration. The total length of this identification cannot exceed eight characters.

SIZE.OF.FLEET - This integer variable is the number of aircraft in the fleet being simulated. The output format requires that this variable does not exceed 99999.

USAGE.LIFE - This real variable is the service life in flight hours of the aircraft being simulated. All aircraft in the fleet must have the same service life. The output format requires that this variable does not exceed 9999999.

BEGIN.PRODUCTION - This real variable is the time in flight hours relative to the start of the simulation when the first aircraft enters service. This variable in conjunction with the input variable START.TEST enables the user to start the fatigue test of the element before, after, or at the same time the first aircraft enters service.

PRODUCTION.TIME - This real variable defines the initial aircraft production rate. It is the time in flight hours between aircraft entering service.

2.PRODUCTION.TIME - This real variable defines the second aircraft production rate. It is the time in flight hours between aircraft entering service.



PRCHG - This real variable is the simulation time when the second aircraft production rate takes effect. Note that this time is measured from the time that the first aircraft enters service and not from the start of the simulation.

START.TEST - This real variable is the time in flight hours relative to the start of the simulation when the fatigue test of an element is begun. If no fatigue test is to be conducted, this variable is set to the machine upper limit.

TEST.ACCEL.FACT - This real variable is the fatigue test acceleration factor, that is, the quotient of the equivalent flight hours divided by the fatigue test hours.

C.GROWTH.RATE - This real variable is the corrosion area growth rate in sq. inches per hour for the aircraft being considered. The growth rate for each element in the aircraft is modified by its associated CRK (corrosion resistance rating).

C7 - If a modification is developed because of a fatigue test failure, this real variable is the percentage (expressed as a decimal fraction) of the test life when the inspection frequency is increased.

C28 - This real variable is the percentage (expressed as a decimal fraction) reduction in the remaining fatigue life of an element when corrosion occurs in a stress concentration.

C29 - This real variable is the percentage (expressed as a decimal fraction) reduction in the remaining fatigue life of an element when corrosion occurs outside a stress concentration.

MU.R - This real variable is the mean of the log-normal distribution of the ratio of the actual average fatigue life to the predicted average fatigue life.

SIG.R - This real variable is the standard deviation of the log-normal distribution of the ratio of the actual average fatigue life to the predicted average fatigue life.

A - This real variable is the result of fitting an exponential curve to flight load exceedance data.  $A_{exp}[BS_a]$  is the number of flight loads per hour which exceed the load level  $S_a$ .

B - This real variable is the result of fitting an exponential curve to flight load exceedance data.  $A_{exp}[BS_a]$  is the number of flight loads per hour which exceed the load level  $S_a$ .

LABCD(\*) - This one-dimensional real array of size four contains the initial lengths in flight hours of the inspection intervals of the four levels of scheduled inspections. LABCD(1) corresponds to the A-level interval; LABCD(2) corresponds to the B-level interval; LABCD(3) corresponds to the C-level interval; and LABCD(4) corresponds to the D-level interval.

CABCD(\*) - This one-dimensional real array of size four contains the inspection cost at each level of inspection. CABCD(1) corresponds to the A-level cost; CABCD(2) corresponds to the B-level cost; CABCD(3) corresponds to the C-level cost; and CABCD(4) corresponds to the D-level cost.

SAMPLING - This real variable is the percentage of the fleet to be sampled during a D-level inspection.

LONG.LIST - This alpha variable is read in as "YES" when the long list output option is desired; otherwise, it is read in as "NO."

PCCL - This real variable is the percentage (read in as a decimal fraction) of the element critical crack length at which a crack, which initiated internally, becomes external.

## 2.2 Long List Data

Occasionally in the standard output, elements will appear with unusually long fatigue cracks or early element failures. It is desirable to have a more complete service history of aircraft with these early element failures than that offered by the standard output. This service history is available through what is called the long list option. This output option is accessed by reading in alpha characters "YES" for the aircraft input variable LONG.LIST. After this input, the element description and identification numbers of the aircraft to be tracked are read in. The input variables for the long list option are listed and described below in the order in which they are read in by SAIFE.

NOE - This integer variable is the number of elements to be processed under the long list option.

ELID(\*,\*) - This two-dimensional alpha array of size four by NOE identifies each element to be processed. This identification must appear in the first sixteen columns of the data card and must be identical to the description read into the variable ELEMENT(\*) described in Section IV, 2.3.

NOAC(\*) - This one-dimensional integer array of size NOE is the number of aircraft to be tracked for each corresponding element.

TLID(\*,\*) - This two-dimensional integer array of size NOE by NOAC(\*) contains the identification numbers of the aircraft to be tracked for a particular element.

The above data are read in immediately following the aircraft input data only when the long list option is desired. Normally, the long list option will be used only after a standard run has indicated a problem area. Since the SAIPE program depends on many sequences of random numbers, all elements in the long list run must be in the same sequence as those in the first run. When the long list option is in effect, the standard output is suppressed.

### 2.3 Element Data

The input variables which are unique to each element and must be read in for each element are listed and described below in the order in which they are read in by SAIPE.

ELEMENT(\*) - This one-dimensional alpha array of size four identifies the element being simulated. The total length of this identification cannot exceed sixteen characters.

PREDICTED.LIFE - This real variable is the average element fatigue life in flight hours predicted by analysis. If the actual average fatigue life is known, this variable can be entered as zero. The output format requires that this variable does not exceed 999999.

ACTUAL.AVG.FAT.LIFE - This real variable is the actual average element fatigue life in flight hours determined by fatigue test. If this value is not known before running the simulation, input zero and SAIPE will determine it statistically.

M1.MEAN - This real variable is the average slow crack propagation rate for the element being simulated.

M2.MEAN - This real variable is the average fast crack propagation rate for the element being simulated.

LGHT.TO.FAILURE - This real variable contains the element crack length which corresponds to zero residual strength or level-flight structural failure.

CRIT.CRK.LGT - This real variable is the element critical crack length, that is, the crack length when the crack propagation rate changes from slow to fast.

FSAF.LGT - This real variable is the element fail-safe crack length.

BIRTH.DEFECT.PROBABILITY - This real variable is the probability (expressed as a decimal fraction) that the element has a production defect when the aircraft enters service.

CRR - This integer variable is the element Corrosion Resistance Rating.

SDM.OCCURRENCE.RATE - This real variable is the service damage occurrence rate in occurrences per element per aircraft per flight hour.

LEAD.TIME - This real variable is the time in flight hours between the time when the decision is made to develop a modification and the time when the modification is ready for implementation.

T.FREQ.CHG - This real variable is the factor (expressed as a decimal fraction) by which inspection intervals are decreased because of a fatigue test failure.

S.FREQ.CHG - This real variable is the factor (expressed as a decimal fraction) by which inspection intervals are decreased because of unfavorable service experience.

FREQ.DECREASE - This real variable is the factor (expressed as a decimal fraction) by which inspection intervals are increased because of favorable service experience.

1.PROB - Given that there is fatigue crack initiation, this real variable is the probability that the crack initiates internally. This variable applies to all three allowable fatigue cracks.

C.PROB - Given that there is corrosion initiation, this real variable is the probability that the corrosion initiates internally.

INT.LVL.INS - This alpha variable is the letter designation of the lowest internal inspection level.

EXT.LVL.INS - This alpha variable is the letter designation of the lowest external inspection level.

A.REPAIR.COST - This real variable is the repair cost of a defect detected during an A-level inspection.

B.REPAIR.COST - This real variable is the repair cost of a defect detected during a B-level inspection.

C.REPAIR.COST - This real variable is the repair cost of a defect detected during a C-level inspection.

D.REPAIR.COST - This real variable is the repair cost of a defect detected during a D-level inspection.

MOD.TESTED - This alpha variable indicates whether or not a structural modification is to be fatigue tested. The two acceptable input values are YES or NO.

1ST.TOOLING - This real variable is the tooling cost in the development of the first structural modification.

AD.TOOLING - This real variable is the tooling cost in the development of any additional structural modifications.

1ST.MD.COST - This real variable is the installation cost of the first structural modification.

AD.MD.COST - This real variable is the installation cost of any additional structural modifications.

S.REPAIR.COST - This real variable is the repair cost of a defect detected during a special inspection.

LOCATED.IN.STRESS.CON - This real variable is the probability (expressed as a decimal fraction) that there is corrosion in a stress concentration.

1.CDM.OCCURRENCE.RATE - This real variable is the initial corrosion occurrence rate in occurrences per element per aircraft per flight hour.

2.CDM.OCCURRENCE.RATE - This real variable is the second corrosion occurrence rate in occurrences per element per aircraft per flight hour.

CDM.RATE.CHANGE - This real variable is the aircraft service time in flight hours when the second corrosion occurrence rate takes effect.

## 2.4 Format Specifications

Most of the input data are entered into SAIFF by the free-form read statement. The program has only three formatted read statements.

The aircraft type identification, the alpha array MODEL, is entered under the format specification 2A4. This identification must be contained in the first eight columns of the first card of the Aircraft Input Data. All subsequent data in this section can appear in any columns and on as many cards as desired. All input values must be separated from one another by at least one blank column and a value cannot be continued on the next card.

The element identification alpha array ELID is entered under the format specification 4A4. This identification must be contained in the first sixteen columns of the long list element data card. Subsequent data can appear in any column and on as many cards as desired. When a second element is to be identified, its description must again appear in the first sixteen columns of the data card.

The element identification alpha array ELEMENT is entered under the format specification 4A4. This identification must be contained in the first sixteen columns of the first card of each set of Element Input Data. As in the Aircraft Input Data, all

subsequent data in this section can appear in any columns and on as many cards as desired.

Sample input data consisting of Aircraft Input Data and five sets of Element Input Data are illustrated in Figure 3. The aircraft type identification is HYBRID. The two cards immediately following the aircraft type card contain the Aircraft Input Data. The five elements shown in the sample are WSC-SWB-AFT-0000, WSC-SWB-AFT-0030, WSC-SWB-AFT-0060, WSC-SWB-AFT-0090, and WSC-SWB-AFT-0127. The Element Input Data for each element begins immediately after the element identification on the same card and terminates on the last card before the next element identification. The card following the last set of Element Input Data must contain EOD in the first four columns.

```
HYBRID
500 60000 150 50 100 5000 0 100 .002 .A .20 .40 1.00 .354 .284325
-8.80901 25 200 1000 12000 1440 7040 13563 19691 .25 NO 1.0
WSC-SWB-AFT-0000 745200 0 A.00E-5 1.60E-3 140. 1.371 15.0 1.19E-5 2
13.31E-9 1476 .800 .650 .250 .466 .667 D H 33 7A 144 20A.5 YES
10000 10000 11584 11584 416 .056 9.154E-9 7.526E-8 7400
WSC-SWB-AFT-0030 690000 0 A.00E-5 1.60E-3 140. 1.371 15.0 1.19E-5 3
13.31E-9 1476 .800 .650 .250 .466 .667 D H 33 7A 144 20A.5 YES
10000 10000 11584 11584 416 .056 9.154E-9 7.526E-8 7400
WSC-SWB-AFT-0060 662400 0 A.00E-5 1.60E-3 140. 1.371 15.0 1.19E-5 1
13.31E-9 1476 .800 .650 .250 .466 .667 D H 33 7A 144 20A.5 YES
10000 10000 11584 11584 416 .056 9.154E-9 7.526E-8 7400
WSC-SWB-AFT-0090 593400 0 A.00E-5 1.60E-3 140. 1.371 15.0 1.19E-5 2
13.31E-9 1476 .800 .650 .250 .466 .667 D H 33 7A 144 20A.5 YES
10000 10000 11584 11584 416 .056 9.154E-9 7.526E-8 7400
WSC-SWB-AFT-0127 607200 0 A.00E-5 1.60E-3 140. 1.371 15.0 1.19E-5 3
13.31E-9 1476 .800 .650 .250 .466 .667 D H 33 7A 144 20A.5 YES
10000 10000 11584 11584 416 .056 9.154E-9 7.526E-8 7400
EOD
```

Figure 3. Sample Input Data to Produce Standard Output

Figure 4 illustrates the same data except that the variable LONG.LIST is now entered as "YES." The Long List Input data appears between the Aircraft Input Data and the Element Input Data. The Long List Input causes the program to track aircraft numbers 100, 200, and 300 for the element WSC-SWB-AFT-0000; aircraft numbers 86 and 497 for the element WSC-SWB-AFT-0030; aircraft number 1 for the element WSC-SWB-AFT-0060; aircraft numbers 9, 10, 11, and 12 for the element WSC-SWB-AFT-0090; and aircraft numbers 323, 456, and 472 for the element WSC-SWB-AFT-0127.

### 3. Output

Each element to be simulated by SAIFE is identified by three groups of alpha characters and one group of numeric characters. The alpha characters define the basic element type and general location on the aircraft, and the numeric characters define the specific location of the element by identifying the wing or fuselage station number. For example, an element identified as WNG-STR-CEN-396 would be a wing stringer located midway between the front and rear spars and centered at wing station 396.

```

HYBRID
500 60000 150 50 100 5000 0 100 .002 .8 .20 .40 1.00 .354 .284325
-8.80901 25 200 1000 12000 1440 7080 13563 19691 .25 YES 1.0
5
WSC-SWB-AFT-0000 3 100 200 300
WSC-SWB-AFT-0030 2 86 497
WSC-SWB-AFT-0060 1 1
WSC-SWB-AFT-0090 4 9 10 11 12
WSC-SWB-AFT-0127 3 323 456 472
WSC-SWB-AFT-0000 745200 0 8.00E-5 1.60E-3 140. 1.371 15.0 1.19E-5 2
13.31E-9 1476 .800 .650 .250 .466 .667 D R 33 78 144 208.5 YES
10000 10000 11584 11584 416 .056 9.154E-9 7.526E-8 7400
WSC-SWB-AFT-0030 690000 0 8.00E-5 1.60E-3 140. 1.371 15.0 1.19E-5 3
13.31E-9 1476 .800 .650 .250 .466 .667 D R 33 78 144 208.5 YES
10000 10000 11584 11584 416 .056 9.154E-9 7.526E-8 7400
WSC-SWB-AFT-0060 662400 0 8.00E-5 1.60E-3 140. 1.371 15.0 1.19E-5 1
13.31E-9 1476 .800 .650 .250 .466 .667 D R 33 78 144 208.5 YES
10000 10000 11584 11584 416 .056 9.154E-9 7.526E-8 7400
WSC-SWB-AFT-0090 593400 0 8.00E-5 1.60E-3 140. 1.371 15.0 1.19E-5 2
13.31E-9 1476 .800 .650 .250 .466 .667 D R 33 78 144 208.5 YES
10000 10000 11584 11584 416 .056 9.154E-9 7.526E-8 7400
WSC-SWB-AFT-0127 607200 0 8.00E-5 1.60E-3 140. 1.371 15.0 1.19E-5 3
13.31E-9 1476 .800 .650 .250 .466 .667 D R 33 78 144 208.5 YES
10000 10000 11584 11584 416 .056 9.154E-9 7.526E-8 7400
EOD

```

Figure 4. Sample Input Data to Produce Long List Output

The standard program output consists of two parts. The first part consists of the simulation results for each specific element. This part is printed for each set of Element Input Data. The second part consists of a summary of the first parts for an element type. In the example discussed above, WNG-STR-CEN identifies the element type. Whenever the program encounters a set of Element Input Data in which any single character of the three groups of alpha characters differs from those in the previous set of Element Input Data, a summary is printed.

A third output is available as an option. This long list option gives a more complete service history of certain selected aircraft and is discussed in Section IV,2.2. When the long list option is in effect, the standard output is suppressed.

### 3.1 Element Data

Figure 5 illustrates a sample output for the input shown in Figure 3. The aircraft type identification is the aircraft input array MODEL. The number of aircraft in the fleet is the aircraft input variable SIZE.OF.FLEET. The aircraft service life is the aircraft input variable USAGE.LIFE. The structural element identification is the element input array ELEMENT. The predicted average fatigue life is the element input variable PREDICTED.LIFE. The actual average fatigue life is the element input variable ACTUAL.AVG.FAT.LIFE.

AIRCRAFT TYPE: HYBRID		AIRCRAFT SERVICE LIFE: 60000 HOURS	
NUMBER OF AIRCRAFT IN FLEET: 500			
STRUCTURAL ELEMENT: WSC-SWB-AFT-0000			
PREDICTED AVERAGE FATIGUE LIFE: 745200 HOURS		ACTUAL AVERAGE FATIGUE LIFE: 670108 HOURS	

NUMBER AND TIME TO INITIATION OF AIRCRAFT DEFECTS				
	FIRST CRACK	CORROSION	SERVICE DAMAGE	PRODUCTION DEFECTS
	-----	-----	-----	-----
OCCURRENCES	2	1	2	0
MIN(HRS)	32752	37720	32752	-----
MAX(HRS)	59789	37720	59789	-----
AVG(HRS)	46270	37720	46270	-----

NUMBER AND LENGTH OF CRACKS DETECTED AT EACH LEVEL OF INSPECTION				
	A-LEVEL	B-LEVEL	C-LEVEL	D-LEVEL
	-----	-----	-----	-----
OCCURRENCES	0	1	0	0
MIN(IN)	0.	2.22	0.	0.
MAX(IN)	0.	2.22	0.	0.
AVG(IN)	0.	2.22	0.	0.

NUMBER AND AREA OF CORROSION DEFECTS DETECTED AT EACH LEVEL OF INSPECTION				
	A-LEVEL	B-LEVEL	C-LEVEL	D-LEVEL
	-----	-----	-----	-----
OCCURRENCES	0	0	0	0
MIN(SQ.IN)	0.	0.	0.	0.
MAX(SQ.IN)	0.	0.	0.	0.
AVG(SQ.IN)	0.	0.	0.	0.

INSPECTION INTERVALS(HRS)		RESIDUAL STRENGTH EQUALS FAIL-SAFE STRENGTH	
		AIRCRAFT NO.	FLT. HOURS
		-----	-----
INITIAL	25	1000	12000
2	25	1250	15000
3	25	14750	18750
4	25	1953	23438
5	25	684	8203
6	25	854	10254
7	25	1068	12817

NUMBER OF SPECIAL INSPECTIONS CONDUCTED: 1		STRUCTURAL FAILURES	
NUMBER OF STRUCTURAL MODIFICATIONS: 0		FLT. HOURS	
FINAL ACTUAL AVERAGE MODIFIED FATIGUE LIFE: 670108 HOURS		-----	
NUMBER OF AIRCRAFT MODIFIED IN SERVICE: 0		AIRCRAFT NO.	
		-----	

Figure 5. Sample Standard Output Produced by Input Shown in Figure 3



AIRCRAFT TYPE: HYBRID		AIRCRAFT SERVICE LIFE: 60000 HOURS	
NUMBER OF AIRCRAFT IN FLEET: 500	STRUCTURAL ELEMENT: MSC-SWB-AFT-0030		
PREDICTED AVERAGE FATIGUE LIFE: 690000 HOURS	ACTUAL AVERAGE FATIGUE LIFE: 424261 HOURS		

NUMBER AND TIME TO INITIATION OF AIRCRAFT DEFECTS			
FIRST CRACK	CORROSION	SERVICE DAMAGE	PRODUCTION DEFECTS
-----	-----	-----	-----
1	1	0	0
51286	56541	0	-----
51286	56541	0	-----
51286	56541	0	-----

NUMBER AND LENGTH OF CRACKS DETECTED AT EACH LEVEL OF INSPECTION			
A-LEVEL	B-LEVEL	C-LEVEL	D-LEVEL
-----	-----	-----	-----
0	0	0	0
0.	0.	0.	0.
0.	0.	0.	0.
0.	0.	0.	0.

NUMBER AND AREA OF CORROSION DEFECTS DETECTED AT EACH LEVEL OF INSPECTION			
A-LEVEL	B-LEVEL	C-LEVEL	D-LEVEL
-----	-----	-----	-----
0	0	0	0
0.	0.	0.	0.
0.	0.	0.	0.
0.	0.	0.	0.

INSPECTION INTERVALS(HRS)			
INITIAL	25	200	12000
2	25	200	15000
3	25	200	18750
4	25	200	23438

NUMBER OF SPECIAL INSPECTIONS CONDUCTED: 0	NUMBER OF STRUCTURAL MODIFICATIONS: 0
FINAL ACTUAL AVERAGE MODIFIED FATIGUE LIFE: 424261 HOURS	NUMBER OF AIRCRAFT MODIFIED IN SERVICE: 0

STRUCTURAL FAILURES	RESIDUAL STRENGTH EQUALS FAIL-SAFE STRENGTH
AIRCRAFT NO. -----	AIRCRAFT NO. -----
FLT. HOURS -----	FLT. HOURS -----

AIRCRAFT TYPE: HYBRID  
 NUMBER OF AIRCRAFT IN FLEET: 500 AIRCRAFT SERVICE LIFE: 60000 HOURS  
 STRUCTURAL ELEMENT: MSC-SMB-AFT-0060  
 PREDICTED AVERAGE FATIGUE LIFE: 662400 HOURS ACTUAL AVERAGE FATIGUE LIFE: 547824 HOURS

NUMBER AND TIME TO INITIATION OF AIRCRAFT DEFECTS

	FIRST CRACK	CORROSION	SERVICE DAMAGE	PRODUCTION DEFECTS
	-----	-----	-----	-----
OCCURRENCES	2	2	0	0
MIN(MRS)	41301	21125	0	-----
MAX(MRS)	58746	38060	0	-----
AVG(MRS)	50043	29592	0	-----

NUMBER AND LENGTH OF CRACKS DETECTED AT EACH LEVEL OF INSPECTION

	A-LEVEL	H-LEVEL	C-LEVEL	D-LEVEL	SPECIAL
	-----	-----	-----	-----	-----
OCCURRENCES	0	1	0	0	0
MIN(IN)	0.	1.12	0.	0.	0.
MAX(IN)	0.	1.12	0.	0.	0.
AVG(IN)	0.	1.12	0.	0.	0.

NUMBER AND AREA OF CORROSION DEFECTS DETECTED AT EACH LEVEL OF INSPECTION

	A-LEVEL	H-LEVEL	C-LEVEL	D-LEVEL	SPECIAL
	-----	-----	-----	-----	-----
OCCURRENCES	0	1	0	1	0
MIN(SQ.IN)	0.	6.61	0.	17.63	0.
MAX(SQ.IN)	0.	6.61	0.	17.63	0.
AVG(SQ.IN)	0.	6.61	0.	17.63	0.

INSPECTION INTERVALS(MRS)

INITIAL	25	200	1000	12000
2	25	200	1250	15000
3	25	200	1563	18750
4	25	200	1953	23438
5	25	200	684	8203
6	25	200	854	10254
7	25	200	1068	12817
8	25	200	1335	16022
9	25	200	1669	20027

NUMBER OF SPECIAL INSPECTIONS CONDUCTED: 1  
 NUMBER OF STRUCTURAL MODIFICATIONS: 0  
 FINAL ACTUAL AVERAGE MODIFIED FATIGUE LIFE: 547824 HOURS  
 NUMBER OF AIRCRAFT MODIFIED IN SERVICE: 0

STRUCTURAL FAILURES  
 AIRCRAFT NO. -----  
 RESIDUAL STRENGTH EQUALS FAIL-SAFE STRENGTH  
 AIRCRAFT NO. -----  
 FLT. HOURS -----

AIRCRAFT TYPE: HYBRID  
 NUMBER OF AIRCRAFT IN FLEET: 500 AIRCRAFT SERVICE LIFE: 60000 HOURS  
 STRUCTURAL ELEMENT: MSC-SMB-AFT-0090  
 PREDICTED AVERAGE FATIGUE LIFE: 593400 HOURS ACTUAL AVERAGE FATIGUE LIFE: 539763 HOURS

NUMBER AND TIME TO INITIATION OF AIRCRAFT DEFECTS

	FIRST CRACK	CORROSION	SERVICE DAMAGE	PRODUCTION DEFECTS
OCCURRENCES	0	6	3	0
MIN(MRS)	1603	5542	1603	----
MAX(MRS)	48952	56707	48952	----
AVG(MRS)	33185	25377	23168	----

NUMBER AND LENGTH OF CRACKS DETECTED AT EACH LEVEL OF INSPECTION

	A-LEVEL	B-LEVEL	C-LEVEL	D-LEVEL	SPECIAL
OCCURRENCES	0	2	2	1	0
MIN(IN)	0.	.51	.49	1.05	0.
MAX(IN)	0.	.65	.55	1.05	0.
AVG(IN)	0.	.58	.52	1.05	0.

NUMBER AND AREA OF CORROSION DEFECTS DETECTED AT EACH LEVEL OF INSPECTION

	A-LEVEL	B-LEVEL	C-LEVEL	D-LEVEL	SPECIAL
OCCURRENCES	0	2	0	1	0
MIN(SQ.IN)	0.	1.65	0.	17.84	0.
MAX(SQ.IN)	0.	2.08	0.	17.84	0.
AVG(SQ.IN)	0.	1.86	0.	17.84	0.

INSPECTION INTERVALS(MRS)

INITIAL	25	200	1000	12000
2	25	200	1250	15000
3	25	200	1563	18750
4	25	200	1953	23438
5	25	200	684	8203
6	25	200	854	10254
7	25	200	1068	12817
8	25	200	1335	16022
9	25	200	1669	20027

NUMBER OF SPECIAL INSPECTIONS CONDUCTED: 1  
 NUMBER OF STRUCTURAL MODIFICATIONS: 0  
 FINAL ACTUAL AVERAGE MODIFIED FATIGUE LIFE: 539763 HOURS  
 NUMBER OF AIRCRAFT MODIFIED IN SERVICE: 0

STRUCTURAL FAILURES	RESIDUAL STRENGTH EQUALS FAIL-SAFE STRENGTH
AIRCRAFT NO.	AIRCRAFT NO.
-----	-----
FLT. HOURS	FLT. HOURS
-----	-----

Figure 5 - Continued

AIRCRAFT TYPE: MYBRID  
 NUMBER OF AIRCRAFT IN FLEET: 500 AIRCRAFT SERVICE LIFE: 60000 HOURS  
 STRUCTURAL ELEMENT: MSC-SW8-AFT-0127  
 PREDICTED AVERAGE FATIGUE LIFE: 607200 HOURS ACTUAL AVERAGE FATIGUE LIFE: 976224 HOURS

NUMBER AND TIME TO INITIATION OF AIRCRAFT DEFECTS				
FIRST CRACK	CORROSION	SERVICE DAMAGE	PRODUCTION DEFECTS	
1	3	0	0	
46529	25520	0	----	
46529	59353	0	----	
46529	43195	0	----	

NUMBER AND LENGTH OF CRACKS DETECTED AT EACH LEVEL OF INSPECTION				
A-LEVEL	B-LEVEL	C-LEVEL	D-LEVEL	SPECIAL
0	1	0	0	0
0.	3.33	0.	0.	0.
0.	3.33	0.	0.	0.
0.	3.33	0.	0.	0.

NUMBER AND AREA OF CORROSION DEFECTS DETECTED AT EACH LEVEL OF INSPECTION				
A-LEVEL	B-LEVEL	C-LEVEL	D-LEVEL	SPECIAL
0	1	0	1	0
0.	1.38	0.	2.96	0.
0.	1.38	0.	2.96	0.
0.	1.38	0.	2.96	0.

INSPECTION INTERVALS(MRS)				
INITIAL	25	1000	12000	
2	200	1250	15000	
3	200	1563	18750	
4	200	1953	23438	
5	200	684	8203	
6	200	854	10254	
7	200	1068	12817	

NUMBER OF SPECIAL INSPECTIONS CONDUCTED: 1  
 NUMBER OF STRUCTURAL MODIFICATIONS: 0  
 FINAL ACTUAL AVERAGE MODIFIED FATIGUE LIFE: 976224 HOURS  
 NUMBER OF AIRCRAFT MODIFIED IN SERVICE: 0

STRUCTURAL FAILURES  
 AIRCRAFT NO. \_\_\_\_\_ FLT. HOURS \_\_\_\_\_  
 RESIDUAL STRENGTH EQUALS FAIL-SAFE STRENGTH  
 AIRCRAFT NO. \_\_\_\_\_ FLT. HOURS \_\_\_\_\_

Figure 5 - Continued

# AIRCRAFT TYPE: A7E21D

NUMBER OF AIRCRAFT IN FLEET: 500 AIRCRAFT SERVICE LIFE: 80000 HOURS

## SUMMARY OF STRUCTURAL ELEMENTS: ACC-500-AFT

### NUMBER AND TIME TO INITIATION OF AIRCRAFT DEFECTS

FIRST CRACK	CORROSION	SERVICE DAMAGE	PRODUCTION DEFECTS
12	13	5	0
1603	5582	1483	---
50780	50353	50780	---
49706	33064	32430	---

### NUMBER AND LENGTH OF CRACKS DETECTED AT EACH LEVEL OF INSPECTION

	A-LEVEL	B-LEVEL	C-LEVEL	D-LEVEL	SPECIAL
OCCURRENCES	0	5	2	1	0
MIN(IN)	0.	.51	.60	1.05	0.
MAX(IN)	0.	3.33	.55	1.05	0.
AVG(IN)	0.	1.57	.52	1.05	0.

### NUMBER AND AREA OF CORROSION DEFECTS DETECTED AT EACH LEVEL OF INSPECTION

	A-LEVEL	B-LEVEL	C-LEVEL	D-LEVEL	SPECIAL
OCCURRENCES	0	4	0	3	1
MIN(SQ.IN)	0.	1.38	0.	2.66	35.87
MAX(SQ.IN)	0.	6.61	0.	17.84	35.87
AVG(SQ.IN)	0.	2.93	0.	12.81	35.87

### INSPECTION INTERVALS(HRS)

INITIAL	25	1000	12000
SHORTEST	25	684	8203
LONGEST	25	1953	23458

NUMBER OF SPECIAL INSPECTIONS CONDUCTED: 4

NUMBER OF STRUCTURAL MODIFICATIONS: 0

NUMBER OF AIRCRAFT MODIFIED IN SERVICE: 0

AIRCRAFT NO.	STRUCTURAL FAILURES FLT. HOURS	STA. NO.	RESIDUAL STRENGTH EQUALS FAIL-SAFE STRENGTH AIRCRAFT NO.	FLT. HOURS	STA. NO.
--------------	-----------------------------------	----------	---	------------	----------

Figure 5 - Concluded

The number of occurrences in the fleet and the times to initiation of the four types of aircraft defects considered by SAIFE are displayed next. Whenever there is a fatigue crack initiation and there are no other cracks in the element, a first crack is said to have occurred. A single element can experience more than one first crack in its lifetime by having a crack initiation after a repair. Similarly, there can be more than one occurrence of corrosion and service damage, although the program does not allow more than one corrosion or service damage defect to exist simultaneously in the same element.

Production defects are one-time occurrences unless there is a structural modification installed. These, too, can have production defects. The times to defect initiation are measured from the time when the aircraft enters service for the initial defects and from the time when the aircraft was last repaired for subsequent defects.

Next, the number and lengths of cracks detected in the fleet at each level of inspection are printed. These numbers include second and third crack detections. Following the crack detection output are the number and areas of corrosion defects detected in the fleet at each level of inspection. A history of the inspection interval changes is printed next. Each time that the aircraft service experience indicates that either an interval increase or an interval decrease is needed, the new interval values are printed. Although the number of interval changes allowed in the simulation is unlimited, the output array size limits the number printed to 30.

The number of fleet-wide special inspections performed is printed next. Each special inspection is always preceded by a decrease in inspection intervals. Defects detected during a special inspection can cause an additional decrease in inspection intervals. Next, the number of structural modifications developed is printed. This number includes modifications because of fatigue test failures or aircraft service experience. The final actual average fatigue life is printed next. If there have been no modifications, this number will be the same as that at the top of the page. If there have been modifications, this number is the actual average fatigue life of the most recently developed modification.

Shown next is the number of aircraft modified in service. If the only modification developed was due to a fatigue test failure, this number can be zero if the test life was such that retrofits were not required. If there were more than one modification requiring retrofits, this number can be greater than the size of the fleet. Finally, each time an aircraft experiences structural failure or its residual strength reaches its fail-safe strength, the aircraft number and the number of accrued flight hours are printed. The aircraft number is assigned by its relative time of entry into service. Aircraft No. 1 is the first aircraft to enter service.

### 3.2 Summary Data

The last section of Figure 5 illustrates a sample summary for the element type WSC-SWB-AFT. All the numbers represent a summary of all the specific elements of this type. Except for the shortest and longest inspection intervals, each number in the summary will appear in one of the specific element outputs. As indicated earlier, the number of interval changes allowed in the simulation is unlimited, and the number printed for a specific element is limited to 30. The shortest and longest intervals printed in the summary are determined from the unlimited number of changes occurring in the simulation.

### 3.3 Long List Data

Figure 6 illustrates a sample long list output for the input shown in Figure 4. For each element the long list headings contain the following: aircraft description, number of aircraft in the fleet, aircraft service life, element description, predicted average fatigue life of the element, actual average fatigue life of the element, and the initial inspection intervals. For each aircraft being tracked, the long list option causes selected information to be printed each time the program control passes to certain events and routines. These events and routines along with the information printed are the following:

Event ENTER.SERVICE - Prints aircraft identification number, number of hours from start of simulation, projected flight hours until crack initiations, and the slow and fast crack growth rates.

Routine INSTALL.MODIFICATION - Prints aircraft identification number, flight time on aircraft, flight hours until crack initiations, and the slow and fast crack growth rates.

Event IN.SERVICE.DAMAGE - Prints aircraft identification number and flight time on aircraft.

Event CORROSION - Prints aircraft identification number, flight time on aircraft, revised slow and fast crack growth rates, revised times until crack initiations, and revised time until failure.

Event 1.STRENGTH.REDUCTION - Prints aircraft identification number, flight time on aircraft, and projected flight hours until element failure.

Event 2.STRENGTH.REDUCTION - Prints aircraft identification number, flight time on aircraft, and projected flight hours until element failure.

Event 3.STRENGTH.REDUCTION - Prints aircraft identification number, flight time on aircraft, and projected flight hours until element failure.

Event 1.ITE - Prints aircraft identification number, length of crack, and flight time on aircraft.

Event 2.ITE - Prints aircraft identification number, length of crack, and flight time on aircraft.

Event 3.ITE - Prints aircraft identification number, length of crack, and flight time on aircraft.

Event D.LEVEL.INSPECTION - Prints aircraft identification number and flight time on aircraft. If an inspection interval increase is implemented at this time, revised intervals are also printed.

Routine EXAMINE - For each defect found, prints size of defect, level of inspection, aircraft identification number, and flight time on aircraft.

Event REACH.FAIL.SAFE.LGT - Prints aircraft identification number and flight time on aircraft.

Event FAILURE - Prints aircraft identification number, flight time on aircraft, sum of crack lengths, and element residual strength.

Event RETIRE.FROM.SERVICE - Prints aircraft identification number and flight time on aircraft.

Event REPAIR - Prints aircraft identification number and projected times to crack initiations.

Event INCREASE.INSPECTION.FREQUENCY - Prints revised C-level and D-level inspection intervals.

Event IMMEDIATE.FLEET.INSPECTION - For each defect found, prints type of defect, size of defect, aircraft identification number, and flight time on aircraft.



AIRCRAFT TYPE: HYBRID

NUMBER OF AIRCRAFT IN FLEET: 500      AIRCRAFT SERVICE LIFE: 60000 HOURS

STRUCTURAL ELEMENT: WSC-SWB-AFT-0000

PREDICTED AVERAGE FATIGUE LIFE: 745200 HOURS      ACTUAL AVERAGE FATIGUE LIFE: 670104 HOURS

INITIAL INSPECTION INTERVALS

A-LEVEL	25 HOURS
B-LEVEL	200 HOURS
C-LEVEL	1000 HOURS
D-LEVEL	12000 HOURS

A/C NO. 100 ENTERS SERVICE 4100 HOURS FROM START OF SIMULATION

1ST CRACK INITIATION PROJECTED AT 655534 FLIGHT HOURS  
 2ND CRACK INITIATION PROJECTED AT 848614 FLIGHT HOURS  
 3RD CRACK INITIATION PROJECTED AT 950200 FLIGHT HOURS  
 SLOW CRACK GROWTH RATE = .000000 INCHES/HOUR  
 FAST CRACK GROWTH RATE = .001605 INCHES/HOUR

INSPECTION INTERVAL INCREASE IMPLEMENTED  
 C-LEVEL INTERVAL NOW 1250 HOURS  
 D-LEVEL INTERVAL NOW 15000 HOURS

A/C NO. 200 ENTERS SERVICE 15050 HOURS FROM START OF SIMULATION

1ST CRACK INITIATION PROJECTED AT 786644 FLIGHT HOURS  
 2ND CRACK INITIATION PROJECTED AT 1101450 FLIGHT HOURS  
 3RD CRACK INITIATION PROJECTED AT 1312006 FLIGHT HOURS  
 SLOW CRACK GROWTH RATE = .000070 INCHES/HOUR  
 FAST CRACK GROWTH RATE = .001490 INCHES/HOUR

D-LEVEL INSPECTION PERFORMED ON A/C NO. 100 AT 12000 HOURS

A/C NO. 300 ENTERS SERVICE 24050 HOURS FROM START OF SIMULATION

1ST CRACK INITIATION PROJECTED AT 354101 FLIGHT HOURS  
 2ND CRACK INITIATION PROJECTED AT 453140 FLIGHT HOURS  
 3RD CRACK INITIATION PROJECTED AT 1209263 FLIGHT HOURS  
 SLOW CRACK GROWTH RATE = .000090 INCHES/HOUR  
 FAST CRACK GROWTH RATE = .001463 INCHES/HOUR

D-LEVEL INSPECTION PERFORMED ON A/C NO. 200 AT 12000 HOURS

INSPECTION INTERVAL INCREASE IMPLEMENTED  
 C-LEVEL INTERVAL NOW 1543 HOURS  
 D-LEVEL INTERVAL NOW 18750 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO. 100 AT 27000 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO. 300 AT 12000 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO. 200 AT 24000 HOURS

INSPECTION INTERVAL INCREASE IMPLEMENTED  
 C-LEVEL INTERVAL NOW 1953 HOURS  
 D-LEVEL INTERVAL NOW 24000 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO. 100 AT 45750 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO. 300 AT 30750 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO. 200 AT 42750 HOURS

A/C NO. 100 RETIRED FROM SERVICE AT 60000 FLIGHT HOURS

A/C NO. 200 RETIRED FROM SERVICE AT 60000 FLIGHT HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO. 300 AT 54188 HOURS

A/C NO. 300 RETIRED FROM SERVICE AT 60000 FLIGHT HOURS

INSPECTION INTERVAL DECREASE IMPLEMENTED  
 C-LEVEL INTERVAL NOW 684 HOURS  
 D-LEVEL INTERVAL NOW 8203 HOURS

a. Structural Element: WSC-SWB-AFT-0000

Figure 6. Sample Long List Output Produced by Input Shown in Figure 4

ALERT 41 - SPECIAL INSPECTION PERFORMED

INSPECTION INTERVAL INCREASE IMPLEMENTED  
C-LEVEL INTERVAL NOW 850 HOURS  
D-LEVEL INTERVAL NOW 10250 HOURS

INSPECTION INTERVAL INCREASE IMPLEMENTED  
C-LEVEL INTERVAL NOW 1000 HOURS  
D-LEVEL INTERVAL NOW 12017 HOURS

a. Structural Element: WSC-SWB-AFT-0000 (Concluded)

AIRCRAFT TYPE: HYBRID

NUMBER OF AIRCRAFT IN FLEET: 500

AIRCRAFT SERVICE LIFE: 60000 HOURS

STRUCTURAL ELEMENT: WSC-SWB-AFT-0030

PREDICTED AVERAGE FATIGUE LIFE: 60000 HOURS

ACTUAL AVERAGE FATIGUE LIFE: 424261 HOURS

INITIAL INSPECTION INTERVALS

A-LEVEL 25 HOURS  
B-LEVEL 200 HOURS  
C-LEVEL 1000 HOURS  
D-LEVEL 12000 HOURS

A/C NO. 86 ENTERS SERVICE 4400 HOURS FROM START OF SIMULATION

1ST CRACK INITIATION PROJECTED AT 514080 FLIGHT HOURS  
2ND CRACK INITIATION PROJECTED AT 644584 FLIGHT HOURS  
3RD CRACK INITIATION PROJECTED AT 691441 FLIGHT HOURS  
SLOW CRACK GROWTH RATE = .000073 INCHES/HOUR  
FAST CRACK GROWTH RATE = .001468 INCHES/HOUR

INSPECTION INTERVAL INCREASE IMPLEMENTED  
C-LEVEL INTERVAL NOW 1250 HOURS  
D-LEVEL INTERVAL NOW 15000 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO. 86 AT 12000 HOURS

INSPECTION INTERVAL INCREASE IMPLEMENTED  
C-LEVEL INTERVAL NOW 1563 HOURS  
D-LEVEL INTERVAL NOW 18750 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO. 86 AT 27000 HOURS

A/C NO. 497 ENTERS SERVICE 44750 HOURS FROM START OF SIMULATION

1ST CRACK INITIATION PROJECTED AT 514414 FLIGHT HOURS  
2ND CRACK INITIATION PROJECTED AT 548364 FLIGHT HOURS  
3RD CRACK INITIATION PROJECTED AT 663849 FLIGHT HOURS  
SLOW CRACK GROWTH RATE = .000073 INCHES/HOUR  
FAST CRACK GROWTH RATE = .001468 INCHES/HOUR

INSPECTION INTERVAL INCREASE IMPLEMENTED  
C-LEVEL INTERVAL NOW 1953 HOURS  
D-LEVEL INTERVAL NOW 23438 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO. 86 AT 45750 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO. 497 AT 12000 HOURS

A/C NO. 86 RETIRED FROM SERVICE AT 60000 FLIGHT HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO. 497 AT 35438 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO. 497 AT 58875 HOURS

A/C NO. 497 RETIRED FROM SERVICE AT 60000 FLIGHT HOURS

b. Structural Element: WSC-SWB-AFT-0030

Figure 6 - Continued

AIRCRAFT TYPE: HYABO

NUMBER OF AIRCRAFT IN FLEET: 500      AIRCRAFT SERVICE LIFE: 60000 HOURS

STRUCTURAL ELEMENT: WSC-SWB-AFT-0060

PREDICTED AVERAGE FATIGUE LIFE: 662400 HOURS      ACTUAL AVERAGE FATIGUE LIFE: 547424 HOURS

INITIAL INSPECTION INTERVALS

A-LEVEL	25 HOURS
H-LEVEL	200 HOURS
C-LEVEL	1000 HOURS
D-LEVEL	12000 HOURS

A/C NO. 1 ENTERS SERVICE 150 HOURS FROM START OF SIMULATION

1ST CRACK INITIATION PROJECTED AT 379222 FLIGHT HOURS  
 2ND CRACK INITIATION PROJECTED AT 1099412 FLIGHT HOURS  
 3RD CRACK INITIATION PROJECTED AT 1158443 FLIGHT HOURS  
 SLOW CRACK GROWTH RATE = .000089 INCHES/HOUR  
 FAST CRACK GROWTH RATE = .001774 INCHES/HOUR

D-LEVEL INSPECTION PERFORMED ON A/C NO. 1 AT 12000 HOURS

INSPECTION INTERVAL INCREASE IMPLEMENTED  
 C-LEVEL INTERVAL NOW 1250 HOURS  
 D-LEVEL INTERVAL NOW 15000 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO. 1 AT 24000 HOURS

INSPECTION INTERVAL INCREASE IMPLEMENTED  
 C-LEVEL INTERVAL NOW 1563 HOURS  
 D-LEVEL INTERVAL NOW 18750 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO. 1 AT 39000 HOURS

INSPECTION INTERVAL INCREASE IMPLEMENTED  
 C-LEVEL INTERVAL NOW 1953 HOURS  
 D-LEVEL INTERVAL NOW 23438 HOURS

INSPECTION INTERVAL DECREASE IMPLEMENTED  
 C-LEVEL INTERVAL NOW 684 HOURS  
 D-LEVEL INTERVAL NOW 8203 HOURS

FLEET WIDE SPECIAL INSPECTION PERFORMED

D-LEVEL INSPECTION PERFORMED ON A/C NO. 1 AT 53806 HOURS

INSPECTION INTERVAL INCREASE IMPLEMENTED  
 C-LEVEL INTERVAL NOW 854 HOURS  
 D-LEVEL INTERVAL NOW 10254 HOURS

A/C NO. 1 RETIRED FROM SERVICE AT 60000 FLIGHT HOURS

INSPECTION INTERVAL INCREASE IMPLEMENTED  
 C-LEVEL INTERVAL NOW 1068 HOURS  
 D-LEVEL INTERVAL NOW 12817 HOURS

INSPECTION INTERVAL INCREASE IMPLEMENTED  
 C-LEVEL INTERVAL NOW 1335 HOURS  
 D-LEVEL INTERVAL NOW 16022 HOURS

INSPECTION INTERVAL INCREASE IMPLEMENTED  
 C-LEVEL INTERVAL NOW 1669 HOURS  
 D-LEVEL INTERVAL NOW 20027 HOURS

c. Structural Element: WSC-SWB-AFT-0060

Figure 6 - Continued

AIRCRAFT TYPE: MYRIID  
 NUMBER OF AIRCRAFT IN FLEET: 500  
 AIRCRAFT SERVICE LIFE: 60000 HOURS  
 STRUCTURAL ELEMENT: WSC-SWB-AFT-0090  
 PREDICTED AVERAGE FATIGUE LIFE: 543000 HOURS  
 ACTUAL AVERAGE FATIGUE LIFE: 519763 HOURS

# INITIAL INSPECTION INTERVALS

A-LEVEL 25 HOURS  
 B-LEVEL 200 HOURS  
 C-LEVEL 1000 HOURS  
 D-LEVEL 12000 HOURS

A/C NO. 9 ENTERS SERVICE 550 HOURS FROM START OF SIMULATION

1ST CRACK INITIATION PROJECTED AT 479151 FLIGHT HOURS  
 2ND CRACK INITIATION PROJECTED AT 1077594 FLIGHT HOURS  
 3RD CRACK INITIATION PROJECTED AT 1249581 FLIGHT HOURS  
 SLOW CRACK GROWTH RATE = .000083 INCHES/HOUR  
 FAST CRACK GROWTH RATE = .001657 INCHES/HOUR

A/C NO. 10 ENTERS SERVICE 600 HOURS FROM START OF SIMULATION

1ST CRACK INITIATION PROJECTED AT 454008 FLIGHT HOURS  
 2ND CRACK INITIATION PROJECTED AT 704099 FLIGHT HOURS  
 3RD CRACK INITIATION PROJECTED AT 840003 FLIGHT HOURS  
 SLOW CRACK GROWTH RATE = .000074 INCHES/HOUR  
 FAST CRACK GROWTH RATE = .001445 INCHES/HOUR

A/C NO. 11 ENTERS SERVICE 650 HOURS FROM START OF SIMULATION

1ST CRACK INITIATION PROJECTED AT 370220 FLIGHT HOURS  
 2ND CRACK INITIATION PROJECTED AT 898443 FLIGHT HOURS  
 3RD CRACK INITIATION PROJECTED AT 114317 FLIGHT HOURS  
 SLOW CRACK GROWTH RATE = .000089 INCHES/HOUR  
 FAST CRACK GROWTH RATE = .001782 INCHES/HOUR

A/C NO. 12 ENTERS SERVICE 700 HOURS FROM START OF SIMULATION

1ST CRACK INITIATION PROJECTED AT 604293 FLIGHT HOURS  
 2ND CRACK INITIATION PROJECTED AT 661954 FLIGHT HOURS  
 3RD CRACK INITIATION PROJECTED AT 724727 FLIGHT HOURS  
 SLOW CRACK GROWTH RATE = .000074 INCHES/HOUR  
 FAST CRACK GROWTH RATE = .001474 INCHES/HOUR

D-LEVEL INSPECTION PERFORMED ON A/C NO. 9 AT 12000 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO. 10 AT 12000 HOURS

INSPECTION INTERVAL INCREASE IMPLEMENTED

C-LEVEL INTERVAL NOW 1250 HOURS  
 D-LEVEL INTERVAL NOW 15000 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO. 11 AT 12000 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO. 12 AT 12000 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO. 9 AT 24000 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO. 10 AT 27000 HOURS

INSPECTION INTERVAL INCREASE IMPLEMENTED

C-LEVEL INTERVAL NOW 1563 HOURS  
 D-LEVEL INTERVAL NOW 18750 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO. 11 AT 27000 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO. 12 AT 27000 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO. 9 AT 39000 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO. 10 AT 45750 HOURS

INSPECTION INTERVAL INCREASE IMPLEMENTED

C-LEVEL INTERVAL NOW 1953 HOURS  
 D-LEVEL INTERVAL NOW 23418 HOURS

d. Structural Element: WSC-SWB-AFT-0090

Figure 6 - Continued

D-LEVEL INSPECTION PERFORMED ON A/C NO. 11 AT 45750 HOURS  
 D-LEVEL INSPECTION PERFORMED ON A/C NO. 12 AT 45750 HOURS  
 INSPECTION INTERVAL DECREASE IMPLEMENTED  
 C-LEVEL INTERVAL NOW 654 HOURS  
 D-LEVEL INTERVAL NOW 8203 HOURS  
 FIRST AIDE SPECIAL INSPECTION PERFORMED  
 D-LEVEL INSPECTION PERFORMED ON A/C NO. 9 AT 46500 HOURS  
 D-LEVEL INSPECTION PERFORMED ON A/C NO. 10 AT 53453 HOURS  
 INSPECTION INTERVAL INCREASE IMPLEMENTED  
 C-LEVEL INTERVAL NOW 854 HOURS  
 D-LEVEL INTERVAL NOW 10254 HOURS  
 D-LEVEL INSPECTION PERFORMED ON A/C NO. 11 AT 54951 HOURS  
 D-LEVEL INSPECTION PERFORMED ON A/C NO. 12 AT 54951 HOURS  
 D-LEVEL INSPECTION PERFORMED ON A/C NO. 9 AT 54703 HOURS  
 A/C NO. 9 RETIRED FROM SERVICE AT 60000 FLIGHT HOURS  
 A/C NO. 10 RETIRED FROM SERVICE AT 60000 FLIGHT HOURS  
 A/C NO. 11 RETIRED FROM SERVICE AT 60000 FLIGHT HOURS  
 A/C NO. 12 RETIRED FROM SERVICE AT 60000 FLIGHT HOURS  
 INSPECTION INTERVAL INCREASE IMPLEMENTED  
 C-LEVEL INTERVAL NOW 1068 HOURS  
 D-LEVEL INTERVAL NOW 12817 HOURS  
 INSPECTION INTERVAL INCREASE IMPLEMENTED  
 C-LEVEL INTERVAL NOW 1335 HOURS  
 D-LEVEL INTERVAL NOW 14022 HOURS  
 INSPECTION INTERVAL INCREASE IMPLEMENTED  
 C-LEVEL INTERVAL NOW 1669 HOURS  
 D-LEVEL INTERVAL NOW 20027 HOURS

d. Structural Element: WSC-SWB-AFT-0090 (Concluded)

AIRCRAFT TYPE: MYH10  
 NUMBER OF AIRCRAFT IN FLEET: 500 AIRCRAFT SERVICE LIFE: 60000 HOURS  
 STRUCTURAL ELEMENT: WSC-SWB-AFT-0127  
 PREDICTED AVERAGE FATIGUE LIFE: 607200 HOURS ACTUAL AVERAGE FATIGUE LIFE: 976224 HOURS  
 INITIAL INSPECTION INTERVALS  
 A-LEVEL 25 HOURS  
 H-LEVEL 200 HOURS  
 C-LEVEL 1000 HOURS  
 D-LEVEL 12000 HOURS  
 INSPECTION INTERVAL INCREASE IMPLEMENTED  
 C-LEVEL INTERVAL NOW 1250 HOURS  
 D-LEVEL INTERVAL NOW 15000 HOURS  
 A/C NO. 121 ENTERS SERVICE 27350 HOURS FROM START OF SIMULATION  
 1ST CRACK INITIATION PROJECTED AT 1151052 FLIGHT HOURS  
 2ND CRACK INITIATION PROJECTED AT 1562644 FLIGHT HOURS  
 3RD CRACK INITIATION PROJECTED AT 1783574 FLIGHT HOURS  
 SLOW CRACK GROWTH RATE = .000074 INCHES/HOUR  
 FAST CRACK GROWTH RATE = .001487 INCHES/HOUR  
 INSPECTION INTERVAL INCREASE IMPLEMENTED  
 C-LEVEL INTERVAL NOW 1563 HOURS  
 D-LEVEL INTERVAL NOW 18750 HOURS

e. Structural Element: WSC-SWB-AFT-0127

Figure 6 - Continued

D-LEVEL INSPECTION PERFORMED ON A/C NO. 323 AT 12000 HOURS

A/C NO. 456 ENTERS SERVICE 40650 HOURS FROM START OF SIMULATION

1ST CRACK INITIATION PROJECTED AT 448494 FLIGHT HOURS  
2ND CRACK INITIATION PROJECTED AT 1391054 FLIGHT HOURS  
3RD CRACK INITIATION PROJECTED AT 1602494 FLIGHT HOURS  
SLOW CRACK GROWTH RATE = .000094 INCHES/HOUR  
FAST CRACK GROWTH RATE = .001907 INCHES/HOUR

A/C NO. 472 ENTERS SERVICE 42250 HOURS FROM START OF SIMULATION

1ST CRACK INITIATION PROJECTED AT 1436062 FLIGHT HOURS  
2ND CRACK INITIATION PROJECTED AT 2101312 FLIGHT HOURS  
3RD CRACK INITIATION PROJECTED AT 2294671 FLIGHT HOURS  
SLOW CRACK GROWTH RATE = .000066 INCHES/HOUR  
FAST CRACK GROWTH RATE = .001319 INCHES/HOUR

INSPECTION INTERVAL INCREASE IMPLEMENTED

C-LEVEL INTERVAL NOW 1453 HOURS

D-LEVEL INTERVAL NOW 23434 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO. 456 AT 12000 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO. 472 AT 12000 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO. 323 AT 30750 HOURS

INSPECTION INTERVAL DECREASE IMPLEMENTED

C-LEVEL INTERVAL NOW 684 HOURS

D-LEVEL INTERVAL NOW 4203 HOURS

FLEET WIDE SPECIAL INSPECTION PERFORMED

INSPECTION INTERVAL INCREASE IMPLEMENTED

C-LEVEL INTERVAL NOW 854 HOURS

D-LEVEL INTERVAL NOW 10254 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO. 323 AT 45041 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO. 456 AT 31741 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO. 472 AT 30141 HOURS

INSPECTION INTERVAL INCREASE IMPLEMENTED

C-LEVEL INTERVAL NOW 1068 HOURS

D-LEVEL INTERVAL NOW 12817 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO. 323 AT 55295 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO. 456 AT 41995 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO. 472 AT 40395 HOURS

A/C NO. 323 RETIRED FROM SERVICE AT 60000 FLIGHT HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO. 456 AT 54812 HOURS

D-LEVEL INSPECTION PERFORMED ON A/C NO. 472 AT 53212 HOURS

A/C NO. 456 RETIRED FROM SERVICE AT 60000 FLIGHT HOURS

A/C NO. 472 RETIRED FROM SERVICE AT 60000 FLIGHT HOURS

e. Structural Element: WSC-SWB-AFT-0127 (Concluded)

Figure 6 - Concluded

## APPENDIX A

### DETAILED PROGRAM DESCRIPTION

In the following detailed description of the SAIFE program, each event and routine in the program is presented separately. Except for the PREAMBLE, each presentation consists of a description, the definition of the local variables, if any, and a flow chart to illustrate the logic of the event or routine.

# APPENDIX A

## TABLE OF CONTENTS

<u>Section</u>		<u>Page</u>
1.	PREAMBLE . . . . .	33
2.	MAIN . . . . .	40
3.	INITIALIZATION . . . . .	44
4.	SUM.INITIALIZE . . . . .	45
5.	REAL.LIFE . . . . .	46
6.	ENTER.SERVICE . . . . .	48
7.	FATIGUE.LIFE.SCATTER . . . . .	51
8.	INSTALL.MODIFICATION . . . . .	53
9.	IN.SERVICE.DAMAGE . . . . .	55
10.	T.IMPLEMENT.MOD . . . . .	57
11.	PREEDICT.SERVICE.DAMAGE . . . . .	58
12.	PREDICT.CORROSION . . . . .	59
13.	COROSION . . . . .	61
14.	RATE . . . . .	63
15.	1.STRENGTH.REDUCTION . . . . .	65
16.	2.STRENGTH.REDUCTION . . . . .	69
17.	3.STRENGTH.REDUCTION . . . . .	74
18.	1.ITE . . . . .	79
19.	2.ITE . . . . .	80
20.	3.ITE . . . . .	81
21.	INSPECTION.SCHEDULER . . . . .	82
22.	A.LEVEL.INSPECTION . . . . .	84
23.	B.LEVEL.INSPECTION . . . . .	85
24.	C.LEVEL.INSPECTION . . . . .	87
25.	D.LEVEL.INSPECTION . . . . .	89
26.	EXAMINE . . . . .	92
27.	PODD . . . . .	96
28.	CANCEL.SCHEDULED.INSPECTIONS . . . . .	97
29.	REACH.FAIL.SAFE.LGT . . . . .	99
30.	FAILURE . . . . .	100
31.	RETIRE.FROM.SERVICE . . . . .	101
32.	REPAIR . . . . .	102
33.	T.INSPECTION.INCREASE . . . . .	108
34.	INCREASE.INSPECTION.FREQUENCY . . . . .	109
35.	IMMEDIATE.FLEET.INSPECTION . . . . .	110
36.	DECISION.ON.MOD . . . . .	112
37.	IMPLEMENT.MODIFICATION . . . . .	115
38.	DISPLAY.OUTPUT . . . . .	117
39.	SUMMARY . . . . .	118



## 1. PREAMBLE

The PREAMBLE is the definition section of a SIMSCRIPT program. All global variables and global arrays are defined. Temporary entities are defined and tally statistics are identified. Event notices and functions are defined and an event priority order is set. The global real variables are listed and described below in the order in which they appear in the PREAMBLE. Global real variables which are input variables are not included here but can be found in the input section.

### Global Real Variables

FLEET.STR.RED - This variable is the sum of crack lengths found in the fleet since the last inspection frequency change.

1AAFL - This variable is the actual average fatigue life of the element design determined in the MAIN program.

CRRF - Assigned a value in routine INITIALIZATION according to the element corrosion resistance rating, this variable is multiplied by the aircraft corrosion growth rate to give the element corrosion growth rate.

COST.OF.REPAIRS - This variable is the sum of repair costs for the fleet since the last modification.

FIXIT.COST - This variable is the cost of repairing a defect found at a particular inspection level. Its value is set in the inspection events.

CHG.FREQ.TIME - This variable is set equal to TIME.V whenever an inspection interval change is scheduled in the event REPAIR.

1CRKT - Each time a first crack occurs, this variable is set equal to the service time on the aircraft.

1CORT - Each time corrosion occurs, this variable is set equal to the service time on the aircraft.

1SDT - Each time service damage occurs, this variable is set equal to the service time on the aircraft.

ACRKL, BCRKL, CCRKL, DCRKL, SCRKL - Each time a crack is found during an A-level, B-level, C-level, D-level, or Special inspection, the corresponding variable is set equal to the crack length.

ACA, BCA, CCA, DCA, SCA - Each time corrosion is found during an A-level, B-level, C-level, D-level, or Special inspection, the corresponding variable is set equal to the corrosion area.

AIRFRAME.TIME - This variable is the number of flight hours accumulated since the last modification for aircraft no longer in service.

G1CRK - Each time a first crack occurs, this variable is set equal to the service time on the aircraft.

G1COR - Each time corrosion occurs, this variable is set equal to the service time on the aircraft.

G1SD - Each time service damage occurs, this variable is set equal to the service time on the aircraft.

GACRK, GBCRK, GCCRK, GDCRK, GSCRK - Each time a crack is found during an A-level, B-level, C-level, D-level, or Special inspection, the corresponding variable is set equal to the crack length.

GACA, GBCA, GCCA, GDCA, GSCA - Each time corrosion is found during an A-level, B-level, C-level, D-level, or Special inspection, the corresponding variable is set equal to the corrosion area.

CINSL, DINSL - Each time there is an inspection interval change, these variables are set equal to the C-level and D-level intervals, respectively.

KSMP - This variable is set equal to 1.0 in the A-level, B-level, and C-level inspection events and set equal to the D-level sampling percentage in the D-level inspection event.

The global integer variables are listed next. Again, input variables are not included in this list.

#### Global Integer Variables

ID - In each event and routine, this variable is the identification number of the aircraft being processed.

IDCK - This variable is initialized to zero and incremented by one each time an aircraft enters service.

I - This variable is used as a local index or array subscript in different locations in the program.

COUNT.ELEMENT - Each time new element data is read in, this variable is incremented by one.

NICHG - This variable is the number of times that the inspection intervals have changed.

LHTA - This variable is the identification number of the aircraft among the ten high-time aircraft with the fewest flight hours.

1, NUM, OF, RETIRE - This variable is the number of aircraft that have been retired from service.

2, NUM, OF, CRASH - This variable is the number of aircraft which have been removed from service because of structural failure.

ITRN1 - This variable is the numeric identification of the lowest internal level of inspection.

EXT, INSP, LEVEL - This variable is the numeric identification of the lowest external level of inspection.

ILL, ILL - These variables are the numeric identifications of the lowest internal and external levels of inspection, respectively. If either of these variables is less than three, it is set equal to three.

TO, RE, MODIFIED - This variable is the number of aircraft with a pending retrofit modification.

BLN, MODIFIED - This variable is the number of aircraft that have had a current retrofit modification installed.

FDCK - This variable is the number of aircraft in service when a modification is implemented because of a fatigue test failure.

OCR, OCOR, OSDM, OPD - These variables are the number of occurrences of first cracks, corrosion, service damage, and production defects, respectively, for a particular element.

OSCR, OSCO - These variables are the number of cracks and corrosion defects, respectively, detected during a special inspection for a particular element.

NSIC - This variable is the number of special inspections conducted for a particular element.

NSMD - This variable is the number of aircraft modified in service for a particular element.

NSFL - This variable is the number of aircraft experiencing structural failure for a particular element.

NMD - This variable is the number of structural modifications made on a particular element.

NRFS - This variable is the number of aircraft with the residual strength for a particular element reaching the fail-safe strength.

SNRES - This variable is the number of aircraft with the residual strength for a particular element type reaching the fail-safe strength.

J - This variable is used as a local index or array subscript in different locations in the program.

LDA - If the long list option is in effect, this variable is the ascending sequential position of the element being processed among those elements read in under the long list option.

GOLCR, GOCOR, GOSDM - These variables are the number of occurrences of first cracks, corrosion damage, and service damage, respectively, for a particular element type.

GOSCR, GOSCO - These variables are the number of cracks and corrosion defects, respectively, detected during a special inspection for a particular element type.

GOPD - This variable is the number of occurrences of production defects for a particular element type.

SNSIC - This variable is the number of special inspections conducted for a particular element type.

SNMD - This variable is the number of structural modifications made on a particular element type.

SNSMD - This variable is the number of aircraft modified in service for a particular element type.

SNSFL - This variable is the number of aircraft experiencing structural failure for a particular element type.

The real arrays are listed next. Unless otherwise noted, all arrays are 1-dimensional. As before, input arrays are not included in this list.

#### Real Arrays

C.INTERVAL, D.INTERVAL - The elements of these arrays are the current C-level and D-level inspection intervals for each aircraft in the fleet.

ABCD - This array is of size four and contains the most recent intervals for each of the four levels of inspection.

CKREP.TIME - This array is the simulation time of the most recent crack repair for each aircraft.

CORRP.TIME - This array is the simulation time of the most recent corrosion repair for each aircraft.

LAST.SD - This array is the simulation time of the most recent occurrence of service damage for each aircraft.

OCCUR.MOD - This array is the simulation time when the most recent modification was installed for each aircraft.

MSR, MFR - These arrays are the slow and fast crack growth rates, respectively, for each aircraft.

SC, SD - These arrays contain each of the inspection interval changes for the C-level and D-level, respectively.

NRN - This array is the random number selected to calculate the time until structural failure for each aircraft.

MRDD - This array is the simulation time of the most recently detected defect at either a C-level or D-level inspection for each of the ten high-time aircraft.

GCRI - This array is the corrosion multiplying factor for each aircraft.

The following are the integer arrays. Again, unless otherwise noted, all arrays are 1-dimensional and input arrays are not included.

#### Integer Arrays

A1SR, A2SR, A3SR - These arrays contain the event notice identification numbers for each aircraft for the events 1.STRENGTH.REDUCTION, 2.STRENGTH.REDUCTION, and 3.STRENGTH.REDUCTION, respectively.

AF - This array is the event notice identification number for each aircraft for event FAILURE.

AIRPLANE - This array is the temporary entity identification number for each aircraft.

AAL, ABL, ACL, ADL - These arrays are the event notice identification numbers for each aircraft for the events A.LEVEL.INSPECTION, B.LEVEL.INSPECTION, C.LEVEL.INSPECTION, and D.LEVEL.INSPECTION, respectively.

AC, ATII - These arrays are the event notice identification numbers for each aircraft for events COROSION and T.INSPECTION. INCREASE, respectively.

ACID - This array contains the identification numbers of those aircraft experiencing structural failure for a particular element.

OICR, OICO - These arrays are the number of cracks and corrosion defects, respectively, detected at each of the four levels of inspection for a particular element.

SACID - This array contains the identification numbers of those aircraft experiencing structural failure for a particular element type.

GOICR, GOICO - These arrays are the number of cracks and corrosion defects, respectively, detected at each of the four levels of inspection for a particular element type.

HLTIME, ACRFT - This array contains the identification numbers of the ten high-time aircraft.

APID - This array contains the identification numbers of those aircraft with a particular element whose residual strength has reached the fail-safe strength.

SAPID - This array contains the identification numbers of those aircraft with a particular element type whose residual strength has reached the fail-safe strength.

STIM - This array contains the flight hours on each aircraft when the residual strength for a particular element reaches the fail-safe strength.

SSTIM - This array contains the flight hours on each aircraft when the residual strength for a particular element type reaches the fail-safe strength.

FLTHR - This array contains the flight hours on each aircraft when structural failure occurs for a particular element.

SFLTHR - This array contains the flight hours on each aircraft when structural failure occurs for a particular element type.

ARFSL - This array is the event notice identification number for each aircraft for event REACH.FAIL.SAFE.1GT.

A1E, A2E, A3E - These arrays are the event notice identification numbers for each aircraft for events 1.ITE, 2.ITE, and 3.ITE, respectively.

The global alpha arrays are listed next. As before, input arrays are not included in this list.

#### Global Alpha Arrays

1.CR.EXISTS, 2.CR.EXISTS, 3.CR.EXISTS - The elements of these arrays are set equal to "YES" for each aircraft whenever there is a first, second, and third crack initiation, respectively.

CO.EXISTS - This array is set equal to "YES" for each aircraft when it has corrosion initiation.

SD.SCH - This array is set equal to "YES" for each aircraft that has event IN.SERVICE.DAMAGE scheduled.

SSTAN - This array is the station number which identifies each aircraft experiencing the failure of a particular element type.

SELNB - This array is the station number which identifies each aircraft with a particular element type whose residual strength has reached the fail-safe strength.

AIL, FSH - This array is set equal to "YES" for each aircraft when events REACH.FAIL.SAFE.LGT and FAILURE, respectively, are scheduled.

IE1, IE2, IE3 - This array is set equal to "YES" for each aircraft that has events 1.IE, 2.IE, and 3.IE, respectively, scheduled.

TMOD.PENDING - This array is set equal to "YES" for each aircraft that has a modification pending because of a fatigue test failure.

SMOD.PENDING - This array is set equal to "YES" for each aircraft that has a modification pending because of service experience.

INSP.SCH - This array is set equal to "YES" for each aircraft that has inspections below the overhaul level scheduled.

1.INT, 2.INT, 3.INT - These arrays are set equal to "YES" for each aircraft that has a first crack, second crack, or third crack, respectively, initiated internally.

C.INT - This array is set equal to "YES" for each aircraft that has corrosion initiated internally.

The temporary entity definitions and tally statements are self-explanatory. The events, functions, and routines are described in detail in the following sections.

## 2. MAIN

### 2.1 Description

In the MAIN program, space is reserved for all global arrays. The following operations are performed in the order given: all input data is read in; the actual average fatigue life of the element type is calculated; the necessity of a structural modification because of a fatigue test failure is determined; the first event ENTER.SERVICE is scheduled; and the simulation is initiated.

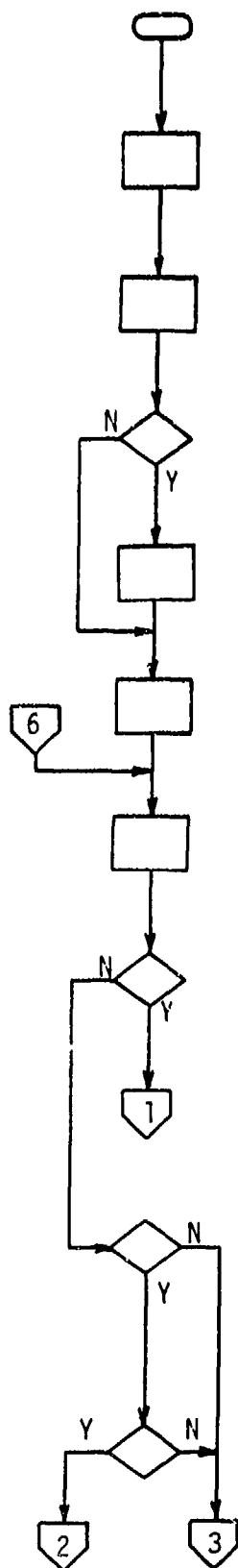
### 2.2 Local Variables

NETS - This real variable is the time in flight hours from when the second production rate goes into effect to when the last aircraft enters service.

SATL - This real variable is the earliest simulation time at which a structural modification because of a fatigue test failure is ready for installation.



## 2.3 Flow Chart



MAIN.

Reserve aircraft arrays.

Read aircraft data.

Does LONG.LIST="YES" ?

Read in elements and aircraft to be tracked.

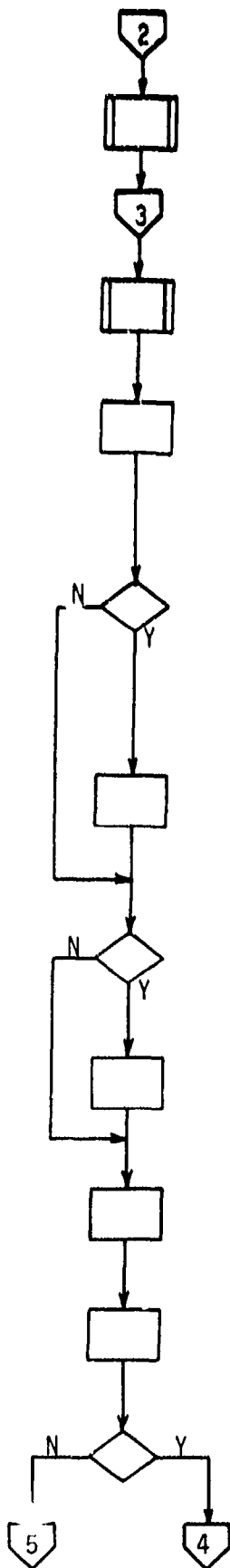
Reserve element arrays.

Read element data.

Does ELEMENT(1)="EOD" ?

Is this element a different element type than previous element?

Does LONG.LIST="NO" ?



Call routine SUMMARY.

Call routine INITIALIZATION.

Calculate actual average fatigue life of element type.

Does fatigue test failure occur at less than twice the aircraft service life?

Schedule a structural modification.

Is long list option in effect?

Print long list headings.

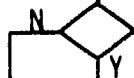
Schedule event ENTER.SERVICE.

Start simulation.

Does LONG.LIST="NO"?



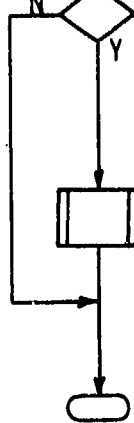
Call routine DISPLAY.OUTPUT.



Does LONG.LIST="NO" ?



Call routine SUMMARY.



END.

### 3. INITIALIZATION

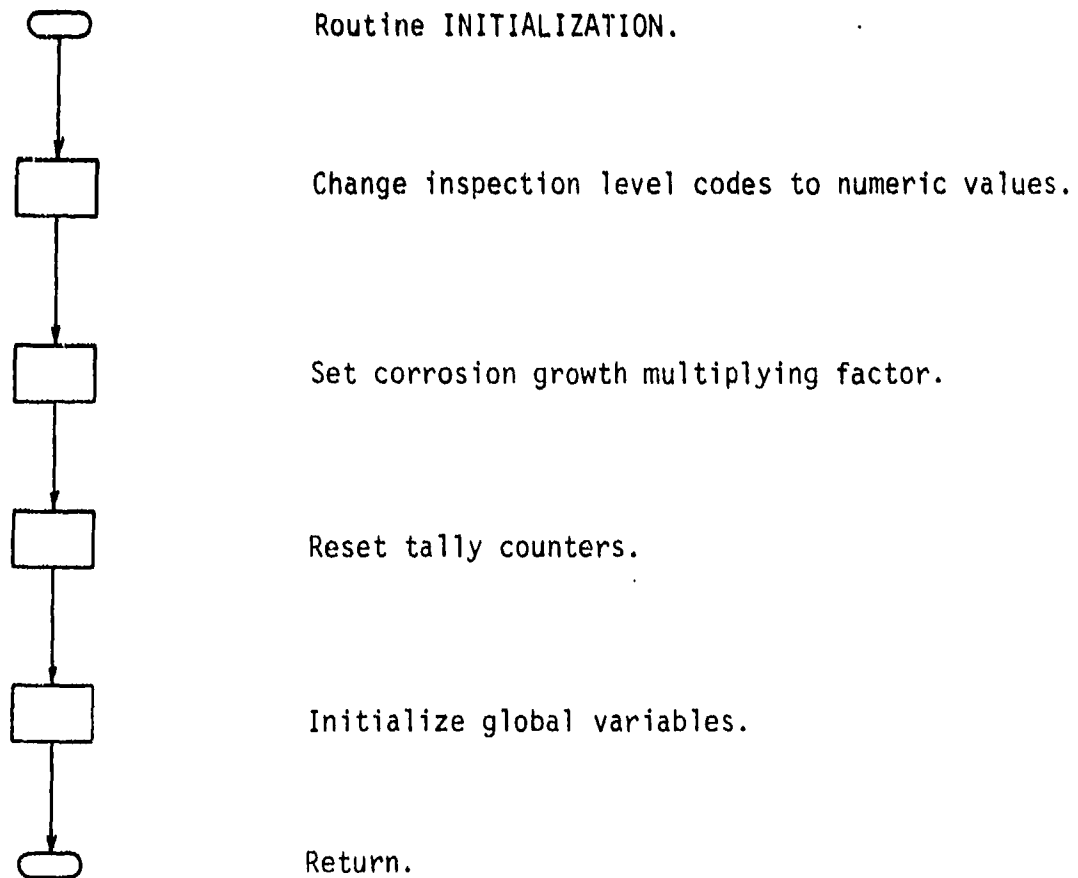
#### 3.1 Description

This routine is called immediately after reading each new set of element input data. This routine changes the inspection level codes to numeric values, sets the corrosion growth multiplying factor based on the corrosion resistance rating, and resets the tally counters. It also initializes all the element global variables which are not part of the input. This routine is called from the MAIN program.

#### 3.2 Local Variables

There are no local variables in this routine.

#### 3.3 Flow Chart



#### 4. SUM.INITIALIZE

##### 4.1 Description

This routine is called each time a new element type is read in. The element type is identified by the first twelve characters of the element identification. This routine initializes the global variables and resets the tally counters. This routine is called from the MAIN program.

##### 4.2 Local Variables

There are no local variables in this routine.

##### 4.3 Flow Chart



Routine SUM.INITIALIZE.

Initialize global variables.

Reset tally counters.

Return.

## 5. REAL.LIFE

### 5.1 Description

This routine accepts (1) the predicted average fatigue life of a particular element design and (2) the mean and standard deviation of the log-normal distribution of the ratio of the actual average fatigue life to the predicted average fatigue life. A random selection is made from the distribution and multiplied by the predicted average fatigue life. The resulting actual average fatigue life is returned to the calling routine. REAL.LIFE can be called from the MAIN program and events IMPLEMENT.MODIFICATION and T.IMPLEMENT.MOD.

### 5.2 Local Variables

MEAN - This real variable, whose value is passed from the calling routine, is the mean of the ratio distribution.

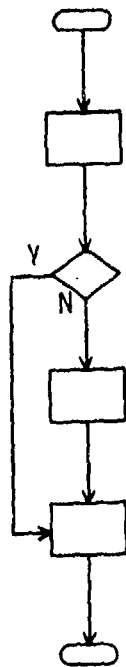
RATIO - This real variable, determined to be log-normally distributed, is the ratio of the actual fatigue life of an element design to its predicted fatigue life.

STD.DEV - This real variable, whose value is passed from the calling routine, is the standard deviation of the ratio distribution.

PDL - This real variable is the design predicted average fatigue life passed from the calling routine.

RFL - This real variable is the element actual average fatigue life which is returned to the calling routine.

### 5.3 Flow Chart



Routine REAL.LIFE.

Randomly select RATIO from distribution.

Is RATIO within allowable limits?

Set RATIO to limit.

Calculate actual average fatigue life.

Return with actual average fatigue life.

## 6. ENTER.SERVICE

### 6.1 Description

This event represents the entry into service of a new aircraft. The temporary entity AIRCRAFT is created and identified by the variable AIRPLANE(ID). The entity attributes TAIL.ID and ENTRY.TIME are defined and the AIRPLANE is filed in the set ACTIVE.FLEET. The routine FATIGUE.LIFE.SCATTER is called to determine the times to first, second, and third crack initiations. The slow and fast crack growth rates are calculated. The times to corrosion initiation and service damage are calculated. If either of these times is less than the service life of the aircraft, the corresponding defect is scheduled. If there is a production defect, the time to first crack initiation is replaced by a time drawn from a distribution of times to crack initiation of aircraft with production defects. If the long list option is in effect for each aircraft being tracked, this routine prints the following: (1) aircraft identification and time it enters service, (2) times to crack initiations, and (3) slow and fast crack growth rates. Crack initiations, D-level inspection, and retirement from service are also scheduled. If the present aircraft is not the last aircraft of the fleet, another ENTER.SERVICE is scheduled. This event can only be scheduled in the MAIN program and within itself.

### 6.2 Local Variables

DEFECT.LIFE - This real variable is the time to first crack initiation when the aircraft has a production defect.

HOURS.TO.CORROSION - This real variable is the time to corrosion initiation.

SECOND.LIFE - This real variable is the time to second crack initiation.

STD.SLOW - This real variable is the standard deviation of the distribution of slow crack growth rates.

FIRST.LIFE - This real variable is the time to first crack initiation when the aircraft has no production defect.

OURS.TO.SERVICE.DAMAGE - This real variable is the time to service damage occurrence.

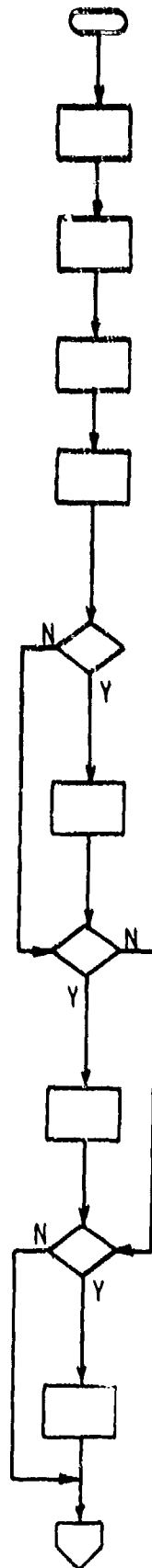
RN - This real variable is a uniformly distributed random number between zero and one.

STD.FAST - This real variable is the standard deviation of the distribution of fast crack growth rates.

THIRD.LIFE - This real variable is the time to third crack initiation.



### 6.3 Flow Chart



Event ENTER SERVICE.

Create an AIRCRAFT and file in ACTIVE.FLEET.

Determine times to first three crack initiations.

Calculate slow and fast crack growth rates.

Determine times to corrosion initiation and service damage.

Is time to corrosion initiation less than aircraft service life?

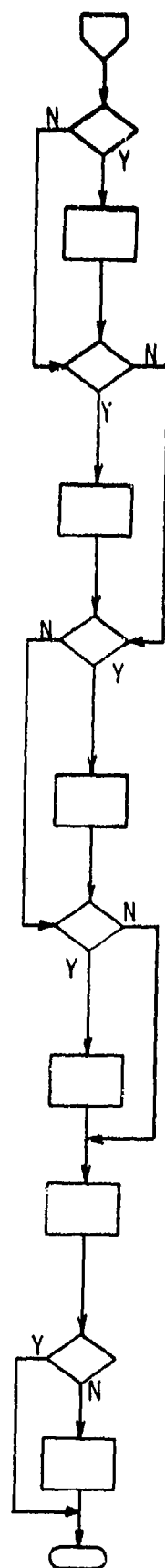
Schedule a corrosion initiation.

Is time to service damage less than aircraft service life?

Schedule a service damage occurrence.

Is random number less than probability of a production defect?

Determine new time to first crack initiation.



Is long list option in effect?

Print long list output if aircraft is one being tracked.

Is time to first crack initiation less than aircraft service life or is corrosion initiation scheduled?

Schedule first crack initiation.

Is time to second crack initiation less than aircraft service life or is corrosion initiation scheduled?

Schedule second crack initiation.

Is time to third crack initiation less than aircraft service life or is corrosion initiation scheduled?

Schedule third crack initiation.

Schedule first D-level inspection and aircraft retirement.

Is this last aircraft to enter service?

Schedule event ENTER.SERVICE.

Return.

## 7. FATIGUE.LIFE.SCATTER

### 7.1 Description

This routine receives the actual average fatigue life of the element design from the calling routine and returns the times to crack initiation of the first three fatigue cracks for the element in a particular aircraft. These times are random selections from a two-parameter Weibull distribution. This routine can be called from routine INSTALL.MODIFICATION and events ENTER.SERVICE and REPAIR.

### 7.2 Local Variables

ALPHA - This real variable is the shape parameter of the fatigue life distribution.

FIRST.LIFE - This real variable is the time to first crack initiation. This time is returned to the calling routine.

N - This integer variable, passed from the calling routine, identifies the random number stream to be used.

RN - This real variable is a uniformly distributed random number.

THIRD.LIFE - This real variable is the time to third crack initiation. This time is returned to the calling routine.

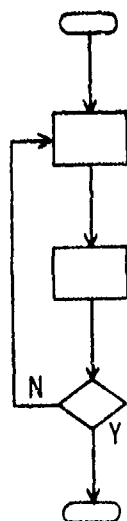
BETA - This real variable is the scale parameter of the fatigue life distribution.

LIFE - This real array of length three is used to temporarily hold the times to crack initiation of the three fatigue cracks.

RFL - This real variable is the element actual average fatigue life passed from the calling routine.

SECOND.LIFE - This real variable is the time to second crack initiation. This time is returned to the calling routine.

### 7.3 Flow Chart



Routine FATIGUE LIFE SCATTER.

Draw uniformly distributed random number.

Calculate time to crack initiation.

Times calculated for these racks?

Return.

## 8. INSTALL.MODIFICATION

### 8.1 Description

This routine represents the installation of a structural modification caused by a fatigue test failure or by aircraft service experience. The modification is installed during a repair or a D-level inspection. All previously scheduled defect initiations are cancelled, and new times to defect initiations are calculated for each aircraft when it is modified. This routine can be called from the events REPAIR and D.LEVEL.INSPECTION.

### 8.2 Local Variables

DEFECT.LIFE - This real variable is the time to crack initiation drawn from a distribution of fatigue lives of elements having production defects.

HOURS.TO.CORROSION - This real variable contains the value returned by routine PREDICT.CORROSION.

RST - This real variable is the remaining service time to retirement of the aircraft being considered.

STD.FAST - This real variable is the standard deviation of the distribution of fast crack growth rates.

THIRD.LIFE - This real variable is the time to third crack initiation returned by routine FATIGUE.LIFE.SCATTER.

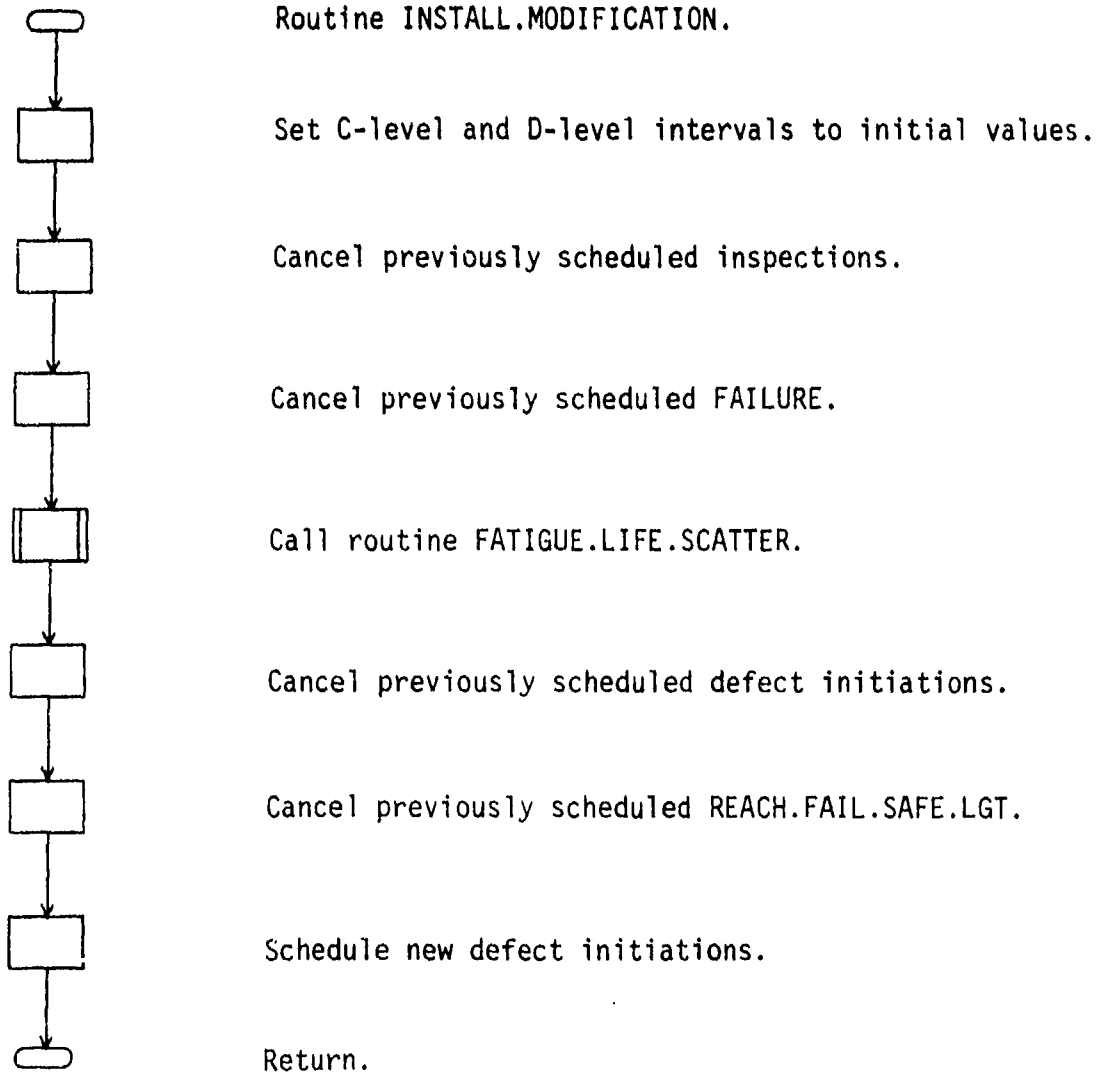
FIRST.LIFE - This real variable is the time to first crack initiation returned by routine FATIGUE.LIFE.SCATTER.

RN - This real variable is drawn from a uniform distribution of random numbers between 0 and 1.

SECOND.LIFE - This real variable is the time to second crack initiation returned by routine FATIGUE.LIFE.SCATTER.

STD.SLOW - This real variable is the standard deviation of the distribution of slow crack growth rates.

### 8.3 Flow Chart



## 9. IN.SERVICE.DAMAGE

### 9.1 Description

This event represents the occurrence of a service damage defect. This occurrence results in the immediate initiation of the next scheduled crack. A new time to service damage is determined. If the new time is less than the remaining time in service of the aircraft, this event is scheduled once again. This event can be scheduled from within itself or in event ENTER.SERVICE.

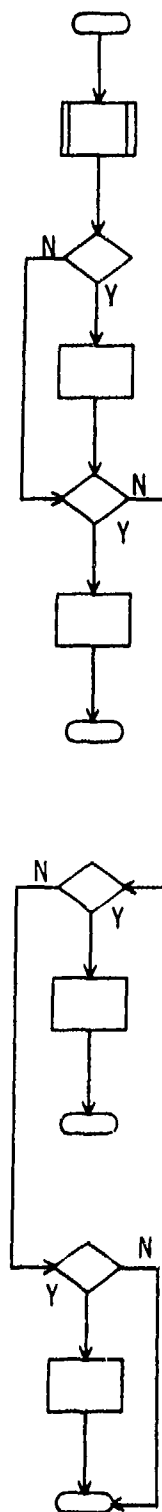
### 9.2 Local Variables

IDSDM - This integer variable is the identification number of the aircraft for which the event was scheduled.

RST - This real variable is the remaining service time to retirement of the aircraft being considered.

OURS.TO.SERVICE.DAMAGE - This real variable is the value returned by routine PREEDICT.SERVICE.DAMAGE.

### 9.3 Flow Chart



Event IN.SERVICE.DAMAGE.

Call routine PREEDICT.SERVICE.DAMAGE.

Is time to service damage less than remaining service time of aircraft?

Schedule event IN.SERVICE.DAMAGE.

Is first crack initiation scheduled?

Reschedule first crack to occur immediately.

Return.

Is second crack initiation scheduled?

Reschedule second crack to occur immediately.

Return.

Is third crack initiation scheduled?

Reschedule third crack to occur immediately.

Return.



## 10. T.IMPLEMENT.MOD

### 10.1 Description

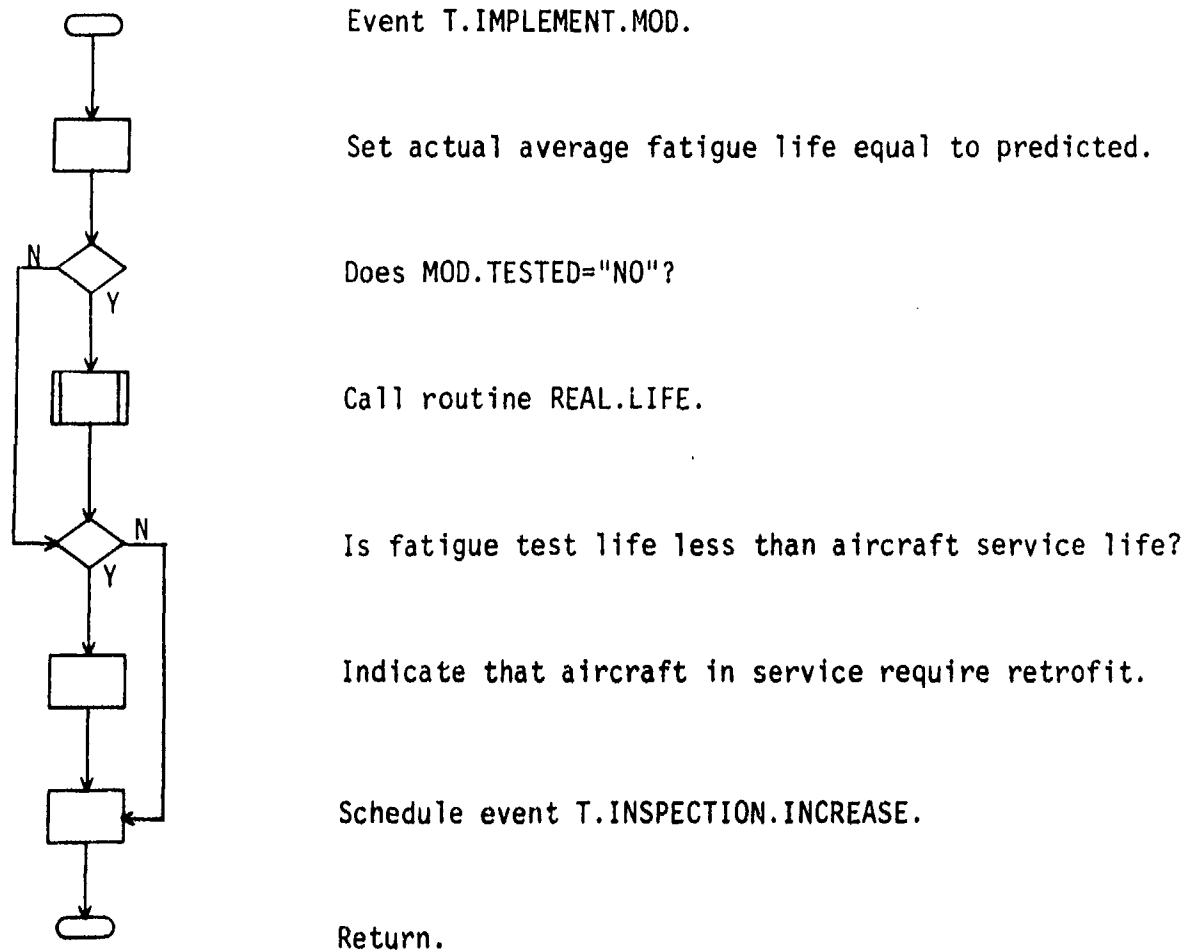
This event represents the development of a structural modification because of a fatigue test failure. This event determines the actual average fatigue life of the modification and schedules an increase in inspection frequencies at some percentage of the fatigue test life. This event is scheduled in the MAIN program.

### 10.2 Local Variables

NSIG - This real variable is  $(SIG.R)(.85)$  and is based on the assumption that a modification usually improves the actual average fatigue life of a particular design.

NMU - This real variable is  $MU.R+0.15(1.0-MU.R)$  and is also based on the foregoing assumption for NSIG.

### 10.3 Flow Chart



## 11. PREEDICT.SERVICE.DAMAGE

### 11.1 Description

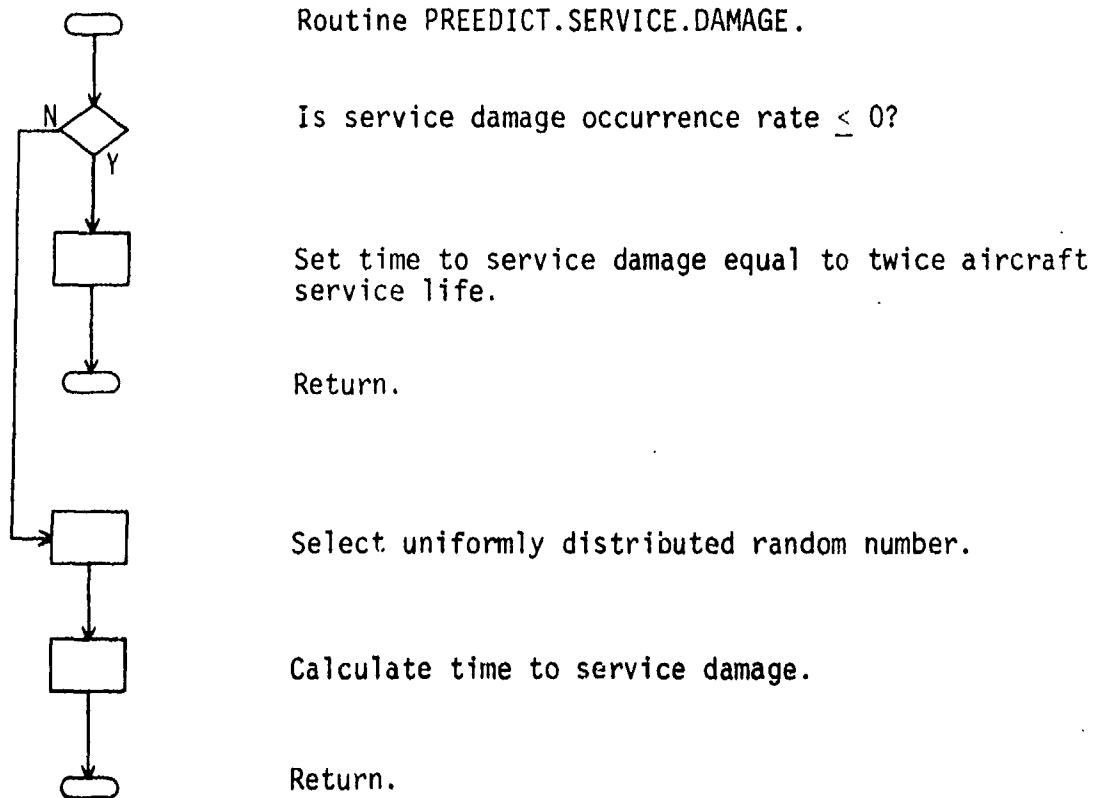
This routine generates the time to service damage occurrence for a given aircraft from a constant service damage occurrence rate. If the service damage occurrence rate is zero in the input, the routine sets the time to service damage occurrence as twice that of the aircraft service life. This routine can be called from events ENTER.SERVICE and IN.SERVICE.DAMAGE.

### 11.2 Local Variables

OURS.TO.SERVICE.DAMAGE - This real variable is the time to service damage occurrence in flight hours. This time is returned by the routine.

RN - This real variable is a uniformly distributed random number between 0 and 1.

### 11.3 Flow Chart



## 12. PREDICT.CORROSION

### 12.1 Description

This routine generates time to corrosion initiation for a given aircraft from a time-dependent occurrence rate approximated by two constant rates. The first constant occurrence rate, the second constant occurrence rate, and the service time on the aircraft when the second rate goes into effect are all input variables. This routine can be called from the routine INSTALL, MODIFICATION and events ENTER.SERVICE and REPAIR.

### 12.2 Local Variables

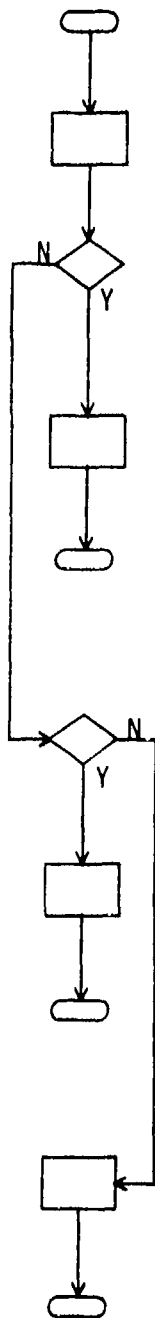
CRCT - This real variable is the remaining time in flight hours until the second corrosion occurrence rate goes into effect. This variable can be negative indicating that the second rate is already in effect.

LD - This real variable is used to hold an intermediate value during the calculation of time to corrosion initiation. The calculation uses a combination of both corrosion occurrence rates.

HOURS.TO.CORROSION - This real variable is the flight time until corrosion initiation. This time is returned to the calling routine.

RN - This real available is a uniformly distributed random number.

### 12.3 Flow Chart



Routine PREDICT.CORROSION.

Select uniformly distributed random number.

Is time on aircraft greater than time at which second occurrence rate goes into effect?

Calculate time to corrosion initiation using second rate.

Return.

Is selected random number > random number required to yield corrosion initiation at time of occurrence rate change?

Calculate time to corrosion initiation using first rate.

Return.

Calculate time to corrosion initiation using combination of both occurrence rates.

Return.

### 13. COROSION

#### 13.1 Description

This event represents the initiation of a corrosion defect. The remaining time to crack initiation of all scheduled cracks is reduced by a corrosion damage factor. If either of the events FAILURE or REACH.FAIL.SAFE.LGT is scheduled, its remaining time until occurrence is also reduced by the corrosion damage factor. This event can be scheduled in the routine INSTALL. MODIFICATION and in events ENTER.SERVICE and REPAIR.

#### 13.2 Local Variables

CDM.MULTIPLYING.FACTOR - This real variable is the factor which when multiplied by the remaining time to crack initiation accounts for the shortening effect of corrosion on fatigue lives.

NFTM - If a FAILURE has been scheduled, this real variable is the remaining time until its occurrence.

REMAINING.LIFE - This real variable is the remaining time until a scheduled crack initiation.

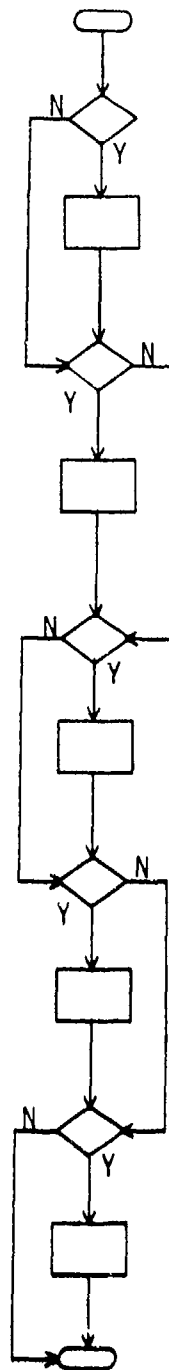
TRT - If a REACH.FAIL.SAFE.LGT has been scheduled, this real variable is the remaining time until its occurrence multiplied by the corrosion damage factor.

IDCO - This integer variable contains the identification number of the aircraft for which the event CORROSION was scheduled.

REDUCED.REMAINING.LIFE - This real variable is the REMAINING.LIFE multiplied by the corrosion damage factor.

RST - This real variable is the remaining service time of the aircraft under consideration.

### 13.3 Flow Chart



Event COROSION .

Is event REACH.FAIL.SAFE.LGT scheduled?

Reduce remaining time to occurrence of REACH.FAIL.SAFE.LGT by corrosion damage factor.

Is event FAILURE scheduled?

Reduce remaining time to occurrence of FAILURE by corrosion damage factor.

Is first crack initiation scheduled?

Reduce remaining time to first crack initiation by corrosion damage factor.

Is second crack initiation scheduled?

Reduce remaining time to second crack initiation by corrosion damage factor.

Is third crack initiation scheduled?

Reduce remaining time to third crack initiation by corrosion damage factor.

Return.

## 14. RATE

### 14.1 Description

This routine statistically generates element crack growth rates which reflect variation in material properties and load environment. The growth rates are randomly drawn from a normal distribution which is defined by a mean growth rate and a standard deviation passed from the calling routine. If a random draw yields a negative growth rate, the rate is set equal to the mean growth rate minus four standard deviations. Thus, the user must be sure that the standard deviation is always less than one-fourth of the mean. This routine is defined as a function in the routine PREAMBLE and is used in event ENTER.SERVICE and routine INSTALL.MODIFICATION.

### 14.2 Local Variables

G1 - This real variable is used to hold intermediate values in the calculation of the crack growth rate.

M - This real variable is the mean crack growth rate passed from the calling routine.

RN - This real variable is a uniformly distributed random number between 0 and 1.

S - This real variable is the crack growth rate standard deviation passed from the calling routine.

Z - This real variable is the element crack growth rate returned to the calling routine.

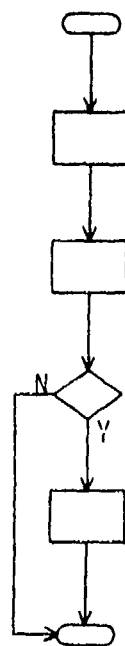
G2 - This real variable is used to hold intermediate values in the calculation of the crack growth rate.

N - This integer variable, passed from the calling routine, identifies the random number stream to be used.

RNI - This real variable, equal to  $1.0 - RN$ , is a uniformly distributed random number between 0 and 1.

W - This real variable is used to hold intermediate values in the calculation of the crack growth rate.

### 14.3 Flow Chart



Routine RATE.

Select uniformly distributed random number.

Calculate crack growth rate.

Is calculated growth rate negative?

Set growth rate equal to mean minus four standard deviations.

Return.



## 15. 1.STRENGTH.REDUCTION

### 15.1 Description

This event represents the initiation of the first crack. If the long list option is in effect, the aircraft identification number and flight hours are printed for those aircraft being tracked. A uniform random number is compared with the probability of internal cracking to determine whether this crack initiates internally. If it does initiate internally, the time until it becomes external is calculated and the event 1.ITE is scheduled. Next, the time to structural failure is calculated by using the three-part residual strength curve described in Vol. II. If this time is less than the remaining service life of the aircraft, the event FAILURE is scheduled. The time until the residual strength of the element reaches the fail-safe strength is calculated, and event REACH.FAIL.SAFE.LGT is scheduled. This event can be scheduled in events ENTER.SERVICE, INSTALL.MODIFICATION, and REPAIR.

### 15.2 Local Variables

ARG - This real variable is used as an intermediate value in the calculation of time until structural failure.

GR2 - This real variable is the fast crack growth rate.

ID1SR - This integer variable is the aircraft identification number.

K1, K11, K13, K4, K9, LG - These real variables are used as intermediate values in the calculation of time until structural failure.

LIST - This real variable is set equal to 1.0 when the long list option is in effect and the aircraft being processed is one of those being tracked.

R2 - This real variable is the strength degradation rate when the crack length is between the critical crack length and the fail-safe length.

SF - This real variable is the element fail-safe strength.

S1 - This real variable is the element residual strength when the crack is at the critical crack length.

TAR - This real variable is the simulation time at which the aircraft being processed retires from service.

T1 - This real variable is the time in flight hours for the crack to grow from its initiation to the critical crack length.

GR1 - This real variable is the element slow crack growth rate.

K10, K12, K2, K8, LGK5 - These real variables are used as intermediate values in the calculation of time until structural failure.

R1 - This real variable is the strength degradation rate from crack initiation to critical crack length.

R3 - This real variable is the strength degradation rate from fail-safe strength to structural failure.

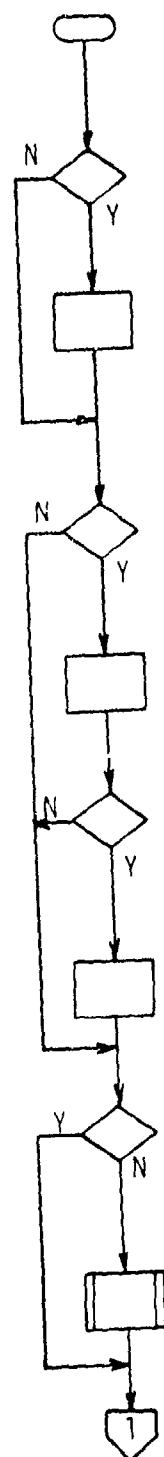
SU - This real variable is the ultimate strength of the element.

T - This real variable is the time in flight hours until a crack initiated internally becomes external.

TTF - This real variable is the time in flight hours until element failure.

T2 - This real variable is the time in flight hours until the fail-safe strength is reached.

### 15.3 Flow Chart



Event 1.STRENGTH.REDUCTION.

Is long list option in effect?

Print ID number and flight hours for aircraft being tracked.

Does crack initiate internally?

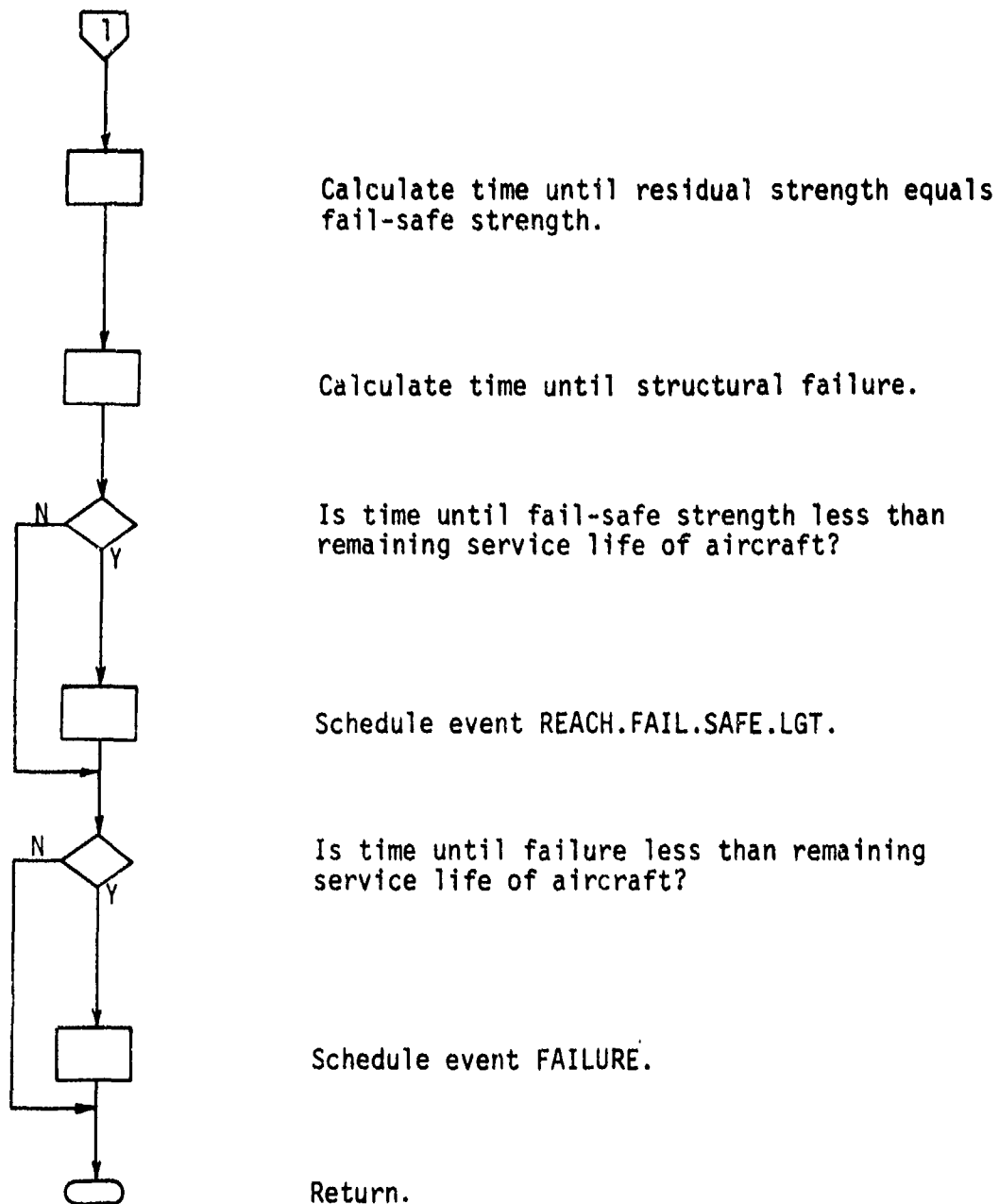
Set 1.INT to "YES".

Is time until crack becomes external less than remaining service life of aircraft?

Schedule event 1.ITE.

Is aircraft presently inspected below overhaul level?

Call routine INSPECTION.SCHEDULER.



## 16. 2.STRENGTH.REDUCTION

### 16.1 Description

This event represents the second crack initiation. If the long list option is in effect, the aircraft identification number and flight hours are printed for those aircraft being tracked. A uniform random number is compared with the probability of internal cracking to determine whether this crack initiates internally. If it does initiate internally, the time until it becomes external is calculated and the event 2.ITE is scheduled. Next, the time to structural failure is calculated by using the same three-part residual strength curve as in event 1.STRENGTH.REDUCTION. However, the crack growth rate is now one calculated by a least-squares fit of points determined from the sum of crack lengths of the two cracks. This calculation is described in Vol. II. If the time until failure is less than the remaining service life of the aircraft, the event FAILURE is scheduled. The time until the residual strength of the element reaches the fail-safe strength is calculated and event REACH.FAIL.SAFE.LGT is scheduled. This event can be scheduled in events ENTER.SERVICE, INSTALL.MODIFICATION, and REPAIR.

### 16.2 Local Variables

ARG - This real variable is used as an intermediate value in the calculation of time until element failure.

CL - This real variable is the crack length when the crack initiated.

GR1 - This real variable is the slow crack growth rate.

K10, K12, K2, K8, LGK5, N - These real variables are used as intermediate values in the calculation of time until element failure.

R2 - This real variable is the strength degradation rate when the composite crack length is between the critical crack length and the fail-safe length.

SF - This real variable is the fail-safe strength of the element.

SU - This real variable is the ultimate strength of the element.

T - This real variable is the time in flight hours until a crack initiated internally becomes external.

TAR - This real variable is the simulation time at which the aircraft being processed retires from service.

TCL - This real variable is the time it takes a single crack to grow from its initiation to the critical crack length.

T1 - This real variable is the time until the first crack reaches its critical crack length.

W, WXS, WY, Y - These real variables are used as intermediate values in the calculation of time until structural failure.

CCL - This real variable is the critical crack length of the element.

DL - This real variable is the length of the first crack at the time of corrosion initiation.

GR2 - This real variable is the fast crack growth rate of the element.

ID2SR - This integer variable is the aircraft identification number.

K1, K11, K13, K4, K9, LG - These real variables are used as intermediate values in the calculation of time until structural failure.

LIST - This real variable is set equal to 1.0 when the long list option is in effect and the aircraft being processed is one of those being tracked.

R1 - This real variable is the strength degradation rate between the time that a first crack initiates until the time that this crack reaches its critical length.

R3 - This real variable is the strength degradation rate from fail-safe strength until structural failure.

SMW - This real variable is used as an intermediate value in the calculation of time until structural failure.

S1 - This real variable is the element residual strength when the first crack is at the critical crack length.

TAC - This real variable is the time of corrosion initiation.

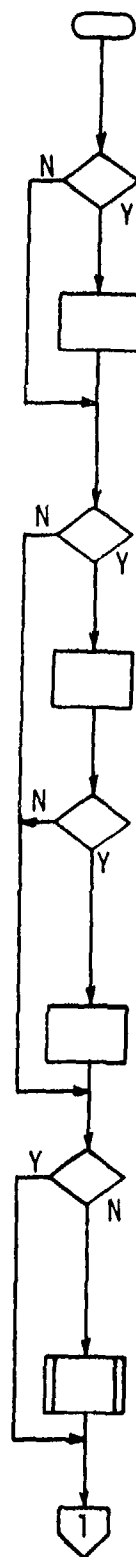
TA1 - This real variable is the time of first crack initiation.

TTF - This real variable is the time in flight hours until structural failure.

T2 - This real variable is the time in flight hours from when an element has a residual strength until when it has a fail-safe strength.

WX, WXY, X - These real variables are used as intermediate values in the calculation of time until structural failure.

### 16.3 Flow Chart



Event 2.STRENGTH.REDUCTION .

Is long list option in effect?

Print ID number and flight hours for aircraft being tracked.

Does crack initiate internally?

Set 2.INT to "YES".

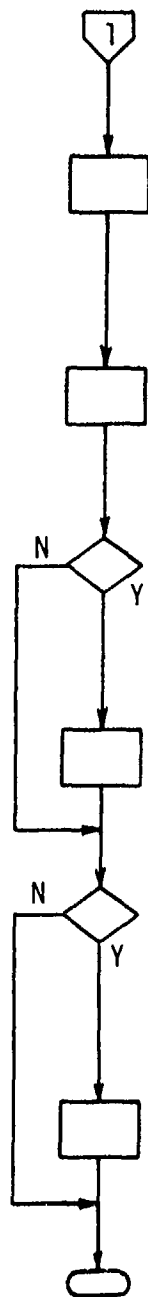
Is time until crack becomes external less than remaining service life of aircraft?

Schedule event 2.ITE.

Is aircraft presently inspected below overhaul level?

Call routine INSPECTION.SCHEDULER.





Calculate time until residual strength equals fail-safe strength.

Calculate time until structural failure.

Is time until fail-safe strength less than remaining service life of aircraft?

Schedule event REACH.FAIL.SAFE.LGT.

Is time until failure less than remaining service life of aircraft?

Schedule event FAILURE.

Return.

## 17. 3.STRENGTH.REDUCTION

### 17.1 Description

This event represents the third crack initiation. If the long list option is in effect, the aircraft identification number and flight hours are printed for those aircraft being tracked. A uniform random number is compared with the probability of internal cracking to determine whether this crack initiates internally. If it does initiate internally, the time until it becomes external is calculated and the event 3.ITE is scheduled. Next, the time to structural failure is calculated by using the same three-part residual strength curve as in event 1.STRENGTH.REDUCTION. However, the crack growth rate is now calculated by a least-squares fit of points determined from the sum of crack lengths of the three cracks. This calculation is described in Vol. II. If the time until failure is less than the remaining service life of the aircraft, the event FAILURE is scheduled. The time until the residual strength of the element reaches the fail-safe strength is calculated and event REACH.FAIL.SAFE.LGT is scheduled. This event can be scheduled in events ENTER.SERVICE, INSTALL.MODIFICATION, and REPAIR.

### 17.2 Local Variables

ARG - This real variable is used as an intermediate value in the calculation of time until structural failure.

CL - This real variable is the crack length when the crack initiated.

GR1 - This real variable is the slow crack growth rate.

K10, K12, K2, K8, LGK5, N - These real variables are used as intermediate values in the calculation of time until structural failure.

R2 - This real variable is the strength degradation rate when the composite crack length is between the critical crack length and the fail-safe length.

SF - This real variable is the fail-safe strength of the element.

SU - This real variable is the ultimate strength of the element.

T - This real variable is the time in flight hours until a crack initiated internally becomes external.

TAR - This real variable is the simulation time at which the aircraft being processed retires from service.

TA2 - This real variable is the time of the second crack initiation.

TTF - This real variable is the time in flight hours until structural failure.

T2 - This real variable is the time in flight hours from when the element has a residual strength until it reaches a fail-safe strength.

WX, WXY, X, Y - These real variables are used as intermediate values in the calculation of time until structural failure.

1CL - This real variable is the crack length of the first crack at third crack initiation.

CCL - This real variable is the critical crack length of the element.

DL - This real variable is the length of the first crack at the time of corrosion initiation.

GR2 - This real variable is the fast crack growth rate of the element.

ID3SR - This integer variable is the aircraft identification number.

K1, K11, K13, K4, K9, LG - These real variables are used as intermediate values in the calculation of time until structural failure.

LIST - This real variable is set equal to 1.0 when the long list option is in effect and the aircraft being processed is one of those being tracked.

R1 - This real variable is the strength degradation rate from when the first crack initiates until it reaches its critical crack length.

R3 - This real variable is the strength degradation rate from fail-safe strength until structural failure.

SMW - This real variable is used as an intermediate value in the calculation of time until structural failure.

S1 - This real variable is the element residual strength when the first crack is at the critical crack length.

TAC - This real variable is the time of corrosion initiation.

TAl - This real variable is the time of first crack initiation.

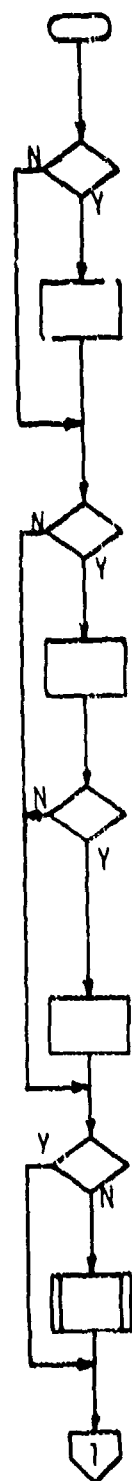
TCL - This real variable is the time it takes a single crack to grow from its initiation to the critical crack length.

T1 - This real variable is the time until the first crack reaches its critical crack length.

W, WXS, WY, XK, Y2 - These real variables are used as intermediate values in the calculation of time until structural failure.

2CL - This real variable is the length of the second crack at the time of the third crack initiation.

### 17.3 Flow Chart



Event 3. STRENGTH REDUCTION .

Is long list option in effect?

Print ID number and flight hours for aircraft being tracked.

Does crack initiate internally?

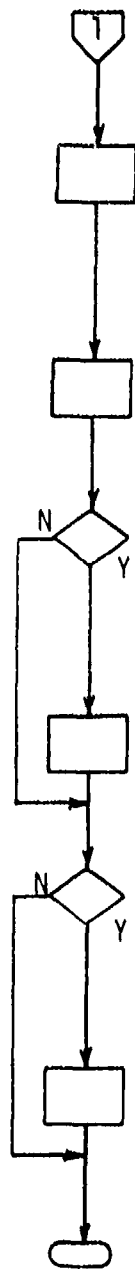
Set 3.INT to "YES".

Is time until crack becomes external less than remaining service life of aircraft?

Schedule event 3.ITE.

Is aircraft presently inspected below overhaul level?

Call routine INSPECTION.SCHEDULER.



Calculate time until residual strength equals fail-safe strength.

Calculate time until structural failure.

Is time until fail-safe strength less than remaining service life of aircraft?

Schedule event REACH.FAIL.SAFE.LGT.

Is time until failure less than remaining service life of aircraft?

Schedule event FAILURE.

Return.

## 18. 1.ITE

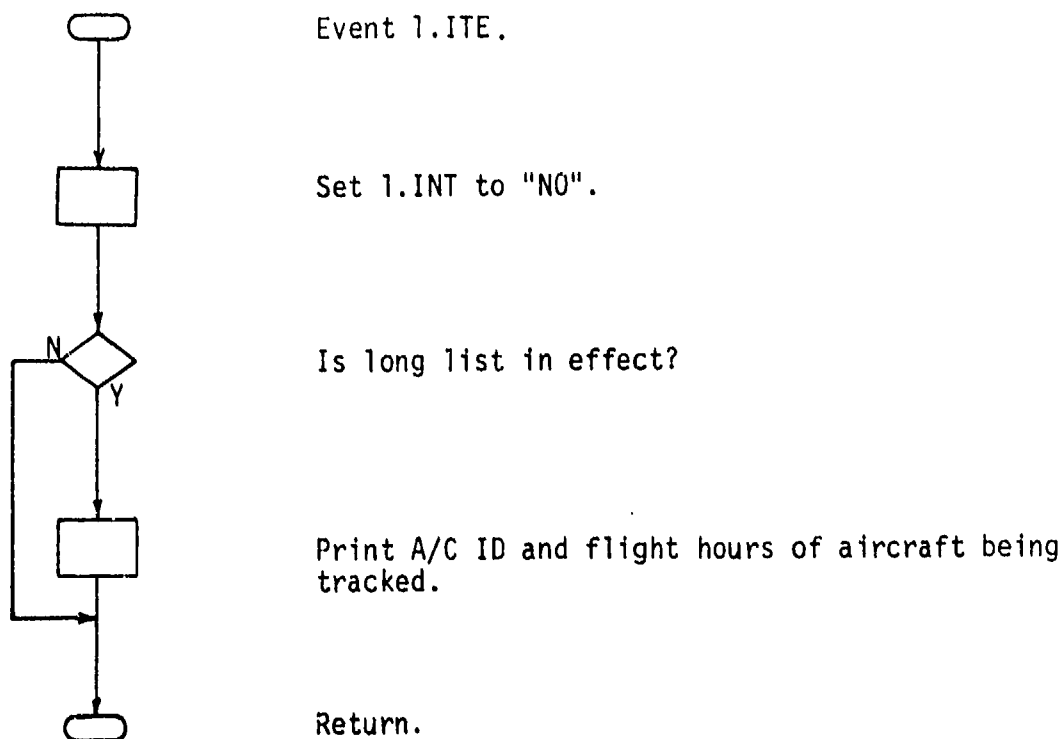
### 18.1 Description

This event represents the time when a first crack which initiated internally becomes external. This time is defined as the time when the element crack length reaches a percentage of the critical crack length. This percentage is an input parameter. At this time the appropriate element of the alpha array 1.INT is changed from "YES" to "NO". If the long list option is in effect, the aircraft identification and flight hours are printed for those aircraft being tracked. This event is scheduled in event 1.STRENGTH.REDUCTION.

### 18.2 Local Variables

ID1E - This integer variable is the aircraft identification number.

### 18.3 Flow Chart



## 19. 2.ITE

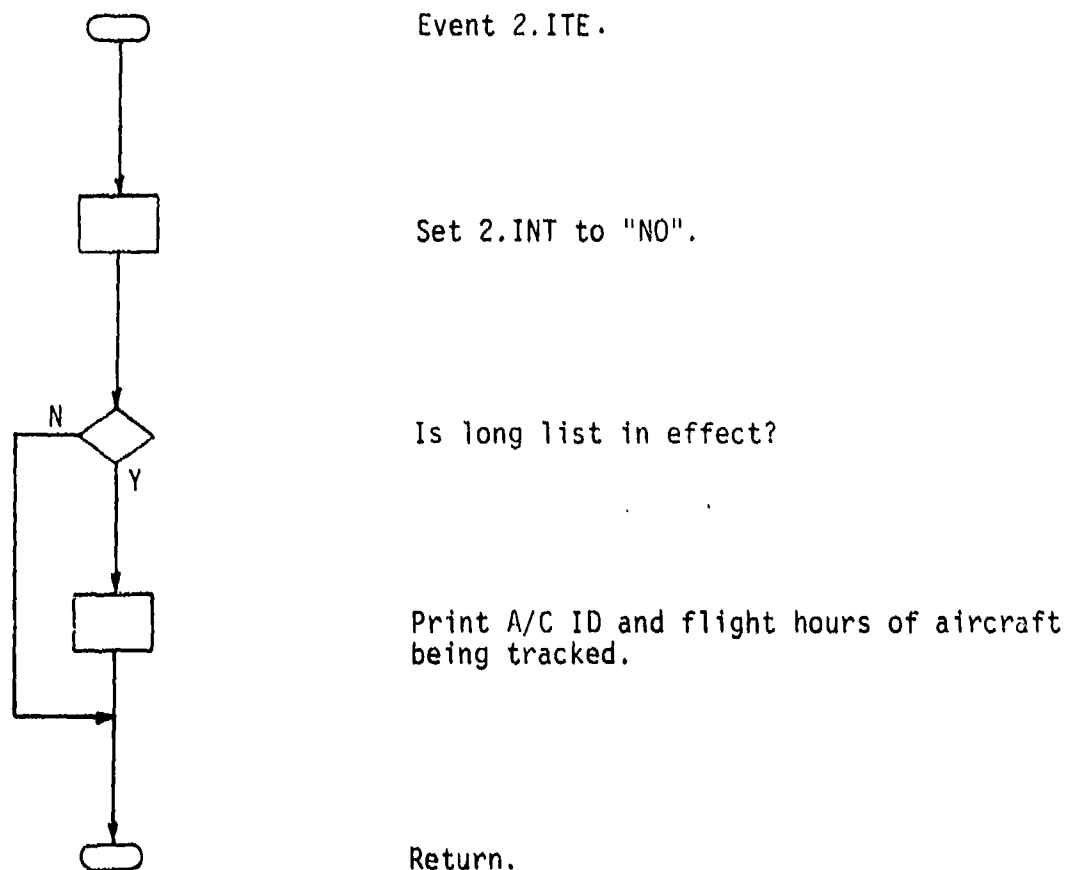
### 19.1 Description

This event represents the time when a second crack which initiated internally becomes external. This time is defined as the time when the element crack length reaches a percentage of the critical crack length. This percentage is an input parameter. At this time the appropriate element of the alpha array 2.INT is changed from "YES" to "NO". If the long list option is in effect, the aircraft identification and flight hours are printed for those aircraft being tracked. This event is scheduled in event 2.STRENGTH.REDUCTION.

### 19.2 Local Variables

ID2E - This integer variable is the aircraft identification number.

### 19.3 Flow Chart





## 20. 3.ITE

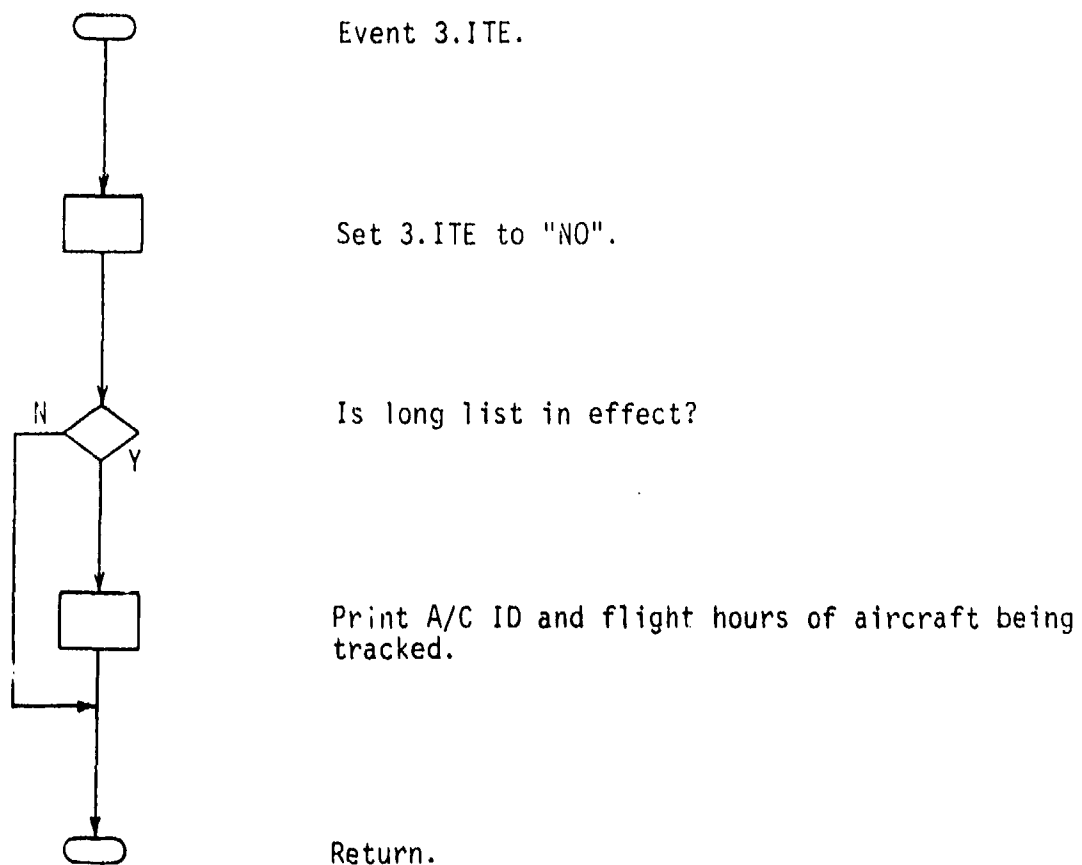
### 20.1 Description

This event represents the time when a first crack which initiated internally becomes external. This time is defined as the time when the element crack length reaches a percentage of the critical crack length. This percentage is an input parameter. At this time the appropriate element of the alpha array 3.INT is changed from "YES" to "NO". If the long list option is in effect, the aircraft identification and flight hours are printed for those aircraft being tracked. This event is scheduled in event 3.STRENGTH.REDUCTION.

### 20.2 Local Variables

ID3E - This integer variable is the aircraft identification number.

### 20.3 Flow Chart



## 21. INSPECTION.SCHEDULER

### 21.1 Description

This routine schedules all inspections below the overhaul level on a given aircraft. To conserve execution time, the inspections are scheduled so that the aircraft is not inspected before the defect reaches its minimum detectable size at each level of inspection. This routine is called from events 1. STRENGTH.REDUCTION and COROSION.

### 21.2 Local Variables

C1 - This real variable is the corrosion growth rate used to calculate the time to the minimum detectable corrosion area.

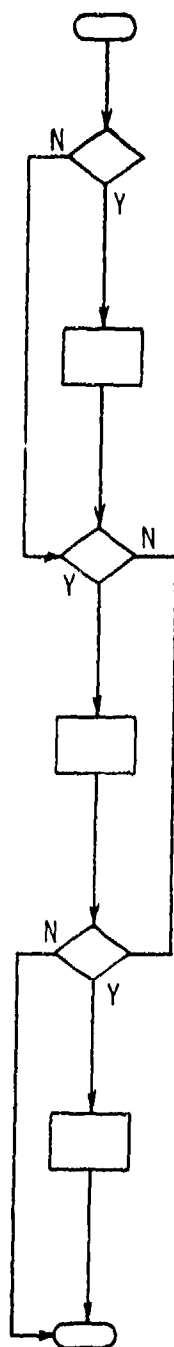
N - This integer variable indicates whether a crack initiation or a corrosion initiation caused this routine to be called.

TML - This real variable is the time to the minimum detectable defect size calculated for each level of inspection.

M1 - This real variable is the crack growth rate used to calculate the time to the minimum detectable crack length.

S.INSPE.AT - This real variable is the simulation time at which the first inspection at each level is scheduled.

### 21.3 Flow Chart



Routine INSPECTION.SCHEDULER.

Is lowest external inspection level less than or equal to A-level?

Calculate time to minimum detectable size.  
Schedule A-level inspection.

Is lowest external inspection level less than or equal to B-level?

Calculate time to minimum detectable size.  
Schedule B-level inspection.

Is lowest external inspection level less than or equal to C-level?

Calculate time to minimum detectable size.  
Schedule C-level inspection.

Return.

## 22. A.LEVEL.INSPECTION

### 22.1 Description

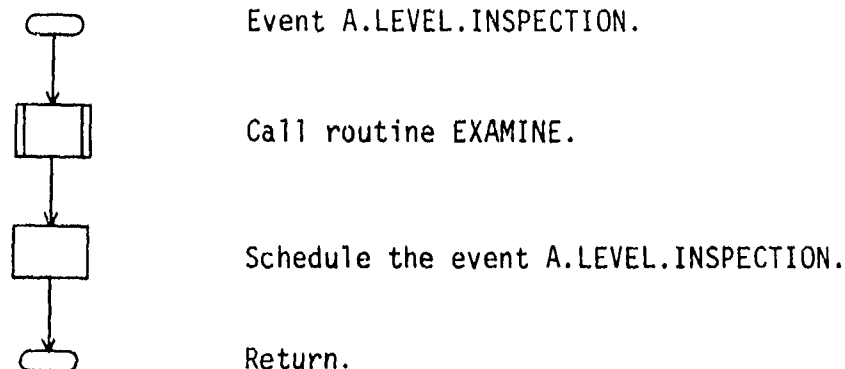
This event represents the performance of an A-level inspection. The constants which define the probability of detection equation at the A-level are passed to the routine EXAMINE which handles the actual inspection calculations for all levels of inspection. This event can be scheduled in the routine INSPECTION. SCHEDULER and the events A.LEVEL.INSPECTION, B.LEVEL.INSPECTION, C.LEVEL.INSPECTION, and D.LEVEL.INSPECTION.

### 22.2 Local Variables

FOUND - Not used in this event, this real variable is returned by routine EXAMINE with a non-zero value whenever a defect is found.

IDA - This integer variable is the identification number of the aircraft being inspected.

### 22.3 Flow Chart



## 23. B.LEVEL.INSPECTION

### 23.1 Description

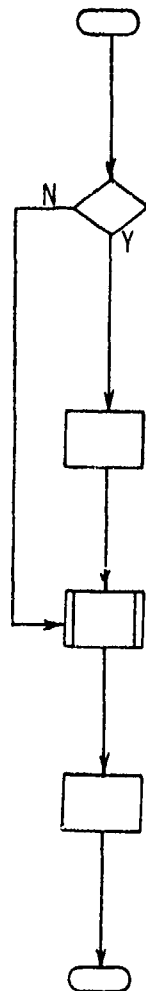
This event represents the performance of a B-level inspection. The constants which define the probability of detection equation at the B-level are passed to the routine EXAMINE. The inspection interval for each inspection level is not necessarily an even multiple of all lower level intervals. If an A-level inspection is scheduled, it is cancelled and rescheduled at present time plus one A-level interval later. This event can be scheduled in the routine INSPECTION.SCHEDULER and the events B.LEVEL.INSPECTION, C.LEVEL.INSPECTION, and D.LEVEL.INSPECTION.

### 23.2 Local Variables

FOUND - Not used in this event, this real variable is returned by routine EXAMINE with a non-zero value whenever a defect is found.

IDB - This integer variable is the identification number of the aircraft being inspected.

### 23.3 Flow Chart



Event B.LEVEL.INSPECTION.

Is lowest external inspection level less than  
or equal to A-level?

Reschedule the A-level inspection.

Call routine EXAMINE.

Schedule the event B.LEVEL.INSPECTION.

Return.

## 24. C.LEVEL.INSPECTION

### 24.1 Description

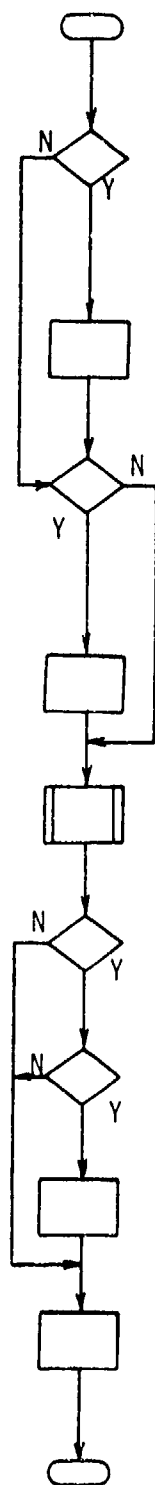
This event represents the performance of a C-level inspection. If there is either an A-level or a B-level inspection currently scheduled, it is cancelled and rescheduled at one A-level interval or B-level interval, respectively, later. The constants which define the probability of defect detection equation at the C-level are passed to the routine EXAMINE. If a crack is detected and the aircraft is one of the ten high-time aircraft, the time of detection is stored. This event can be scheduled in the routine INSPECTION.SCHEDULER and the events C.LEVEL.INSPECTION and D.LEVEL.INSPECTION.

### 24.2 Local Variables

FOUND - This real variable is returned by routine EXAMINE with a value of one whenever a crack is detected.

IDC - This integer variable is the identification number of the aircraft being inspected.

### 24.3 Flow Chart



Event C.LEVEL.INSPECTION.

Is lowest external inspection level less than or equal to B-level?

Reschedule B-level inspection.

Is lowest external inspection level less than or equal to A-level?

Reschedule A-level inspection.

Call routine EXAMINE.

Was crack detected?

Is aircraft one of ten high-time aircraft?

Retain time of detection.

Schedule event C.LEVEL.INSPECTION.

Return.



## 25. D.LEVEL.INSPECTION

### 25.1 Description

This event represents the performance of a D-level inspection. If the long list option is in effect on this element, the identification number and flight hours are printed for each aircraft being tracked. If there are any lower level inspections currently scheduled, they are cancelled and rescheduled at one inspection interval later. The constants which define the probability of defect detection equation at the D-level are passed to the routine EXAMINE. The defect histories of the ten high-time aircraft are now examined. If all of the ten high-time aircraft have gone one D-level interval without any cracks detected at either the C-level or the D-level, then the C-level and D-level inspection intervals are increased by the input factor `FREQ.DECREASE`.

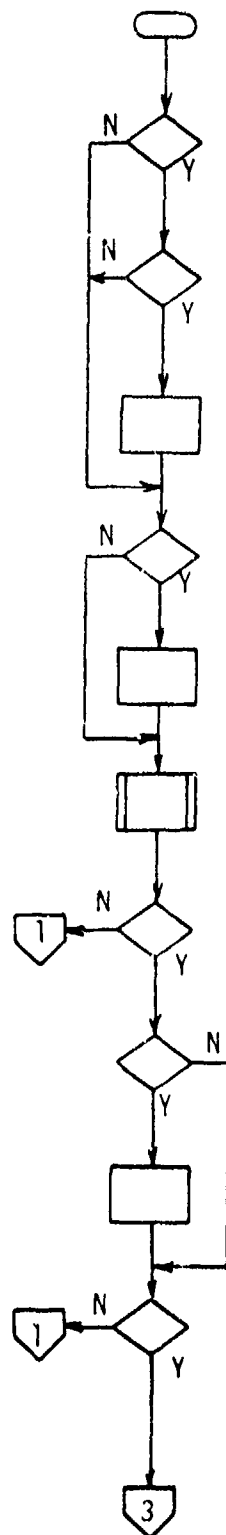
Any pending modifications are installed at this time. This event can be scheduled in the events `ENTER.SERVICE` and `D.LEVEL.INSPECTION`.

### 25.2 Local Variables

FOUND - This real variable is returned by routine EXAMINE with a value of one whenever a crack is detected.

IDD - This integer variable is the identification number of the aircraft being inspected.

### 25.3 Flow Chart



Event D.LEVEL.INSPECTION.

Is long list option in effect on this element?

Is this aircraft being tracked?

Print aircraft number and flight hours.

Are lower level inspections scheduled?

Reschedule lower level inspections.

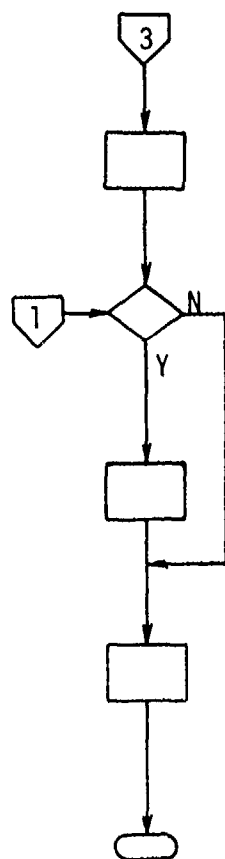
Call routine EXAMINE.

Is aircraft one of ten high-time aircraft?

Was crack detected?

Retain time of detection.

Have all ten high-time aircraft gone one D-level interval without any cracks detected at either C-level or D-level?



Increase C-level and D-level inspection intervals.

Is a modification pending on this aircraft?

Install modification.

Schedule event D.LEVEL.INSPECTION.

Return.

## 26. EXAMINE

### 26.1 Description

This routine performs the numerical comparison which determines whether a defect is detected at each level of inspection. The constants which define the probability of defect detection equation are passed to this routine from the events which represent the different levels of inspection. If the long list option is in effect, each time a defect is detected on one of the aircraft being tracked, the following are printed: the size of the defect, the inspection level, the aircraft identification number, and the number of flight hours on the aircraft. This routine can be called from events A.LEVEL.INSPECTION, B.LEVEL.INSPECTION, C.LEVEL.INSPECTION, and D.LEVEL.INSPECTION.

### 26.2 Local Variables

AREA - This real variable is the calculated area of the corrosion defect.

CL - This real variable is the calculated length of the fatigue cracks.

LIST - This real variable is set equal to 1.0 if the long list option is in effect and the aircraft being inspected is one of those being tracked.

M2 - This real variable is the fast crack growth rate for the aircraft being inspected.

TAC - This real variable is the simulation time of the corrosion initiation.

TA2 - This real variable is the simulation time of the second crack initiation.

XA - This real variable is one of the probability of detection equation constants passed by the calling event.

YA - This real variable is one of the probability of detection equation constants passed by the calling event.

Z - This alpha variable is the level of inspection being performed.

ZL - This real variable is one of the probability of detection equation constants passed by the calling event.

CCL - This real variable is the critical crack length of the element.

FOUND - This real variable is set equal to two whenever corrosion is detected and to one whenever a crack is detected.

M1 - This real variable is the slow crack growth rate for the aircraft being inspected.

N - This integer variable identifies the inspection level and is passed by the calling event.

TA1 - This real variable is the simulation time of the first crack initiation.

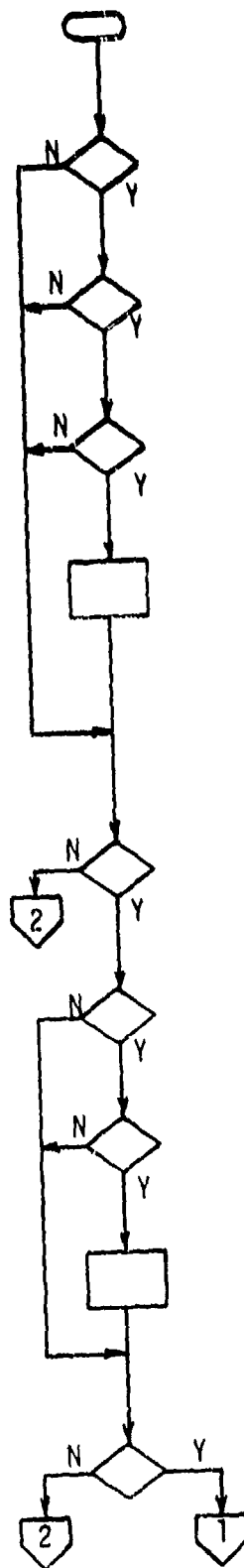
TA3 - This real variable is the simulation time of the third crack initiation.

XL - This real variable is one of the probability of detection equation constants passed by the calling event.

YL - This real variable is one of the probability of detection equation constants passed by the calling event.

ZA - This real variable is one of the probability of detection equation constants passed by the calling event.

## 20.3 Flow Chart



Routine EXAMINE.

Does corrosion exist?

Is the area inspected?

Is random number less than probability of detection?

Set tally counters to defect size.  
Let FOUND=2.

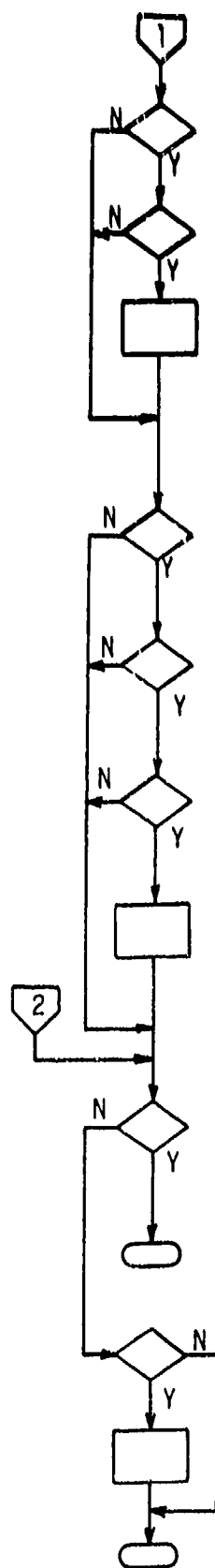
Does first crack exist?

Is the area inspected?

Is random number less than probability of detection?

Set tally counters to defect size.  
Let FOUND=1.

Does second crack exist?



Is the area inspected?

Is random number less than probability of detection?

Set tally counters to defect size.  
Let FOUND=1.

Does third crack exist?

Is the area inspected?

Is random number less than probability of detection?

Set tally counters to defect size.  
Let FOUND=1.

Was routine called from D.LEVEL.INSPECTION?

Return.

Is FOUND greater than zero?

Schedule event REPAIR.

Return.

## 27. PODD

### 27.1 Description

This routine, defined as a function in the PREAMBLE, computes the probability of detecting a crack or corrosion defect of a given size. This probability is returned to the calling routine. PODD is called from routine EXAMINE and event IMMEDIATE. FLEET.INSPECTION.

### 27.2 Local Variables

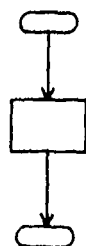
L - This real variable is the size of the defect under consideration.

Y - This real variable is an empirically determined equation constant for each level of inspection.

X - This real variable is the maximum probability of detection at a given inspection level.

Z - This real variable is the minimum defect size detectable at a given inspection level.

### 27.3 Flow Chart



Routine PODD.

Compute probability of detection.

Return.



## 28. CANCEL.SCHEDULED.INSPECTIONS

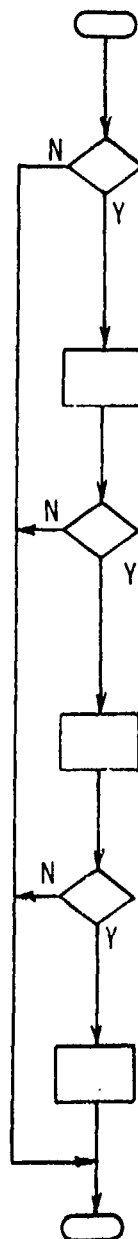
### 28.1 Description

This routine cancels all scheduled inspections below the overhaul level on a given aircraft. Whenever a defect is detected and repaired, it is assumed that all other defects existing on that particular element are also repaired. This routine is called at this time to cancel all subsequent inspections. Also, if an element fails or an aircraft with existing defects is retired, this routine is called to cancel all scheduled inspections. This routine can be called from events FAILURE, RETIRE, FROM.SERVICE, and REPAIR.

### 28.2 Local Variables

There are no local variables in this routine.

### 28.3 Flow Chart



Routine CANCEL.SCHEDULED.INSPECTIONS.

Is lowest level external inspection less than or equal to C-level?

Cancel the C-level inspection.

Is lowest level external inspection less than or equal to B-level?

Cancel the B-level inspection.

Is lowest level external inspection equal to A-level?

Cancel the A-level inspection.

Return.

## 29. REACH.FAIL.SAFE.LGT

### 29.1 Description

This event represents the time when the residual strength of the element has been reduced to the fail-safe strength. The time and aircraft identification number are saved as part of the output. The calculation of the strength reduction is based on the sum of all crack lengths in the element. This event can be scheduled in events 1.STRENGTH.REDUCTION, 2.STRENGTH.REDUCTION, and 3.STRENGTH.REDUCTION.

### 29.2 Local Variables

IDRFS - This integer variable is the identification number of the aircraft being processed.

### 29.3 Flow Chart



Event REACH.FAIL.SAFE.LGT.

Is long list option in effect?

Is this aircraft one of those being tracked?

Print aircraft number and flight hours.

Build output arrays.

Return.

### 30. FAILURE

#### 30.1 Description

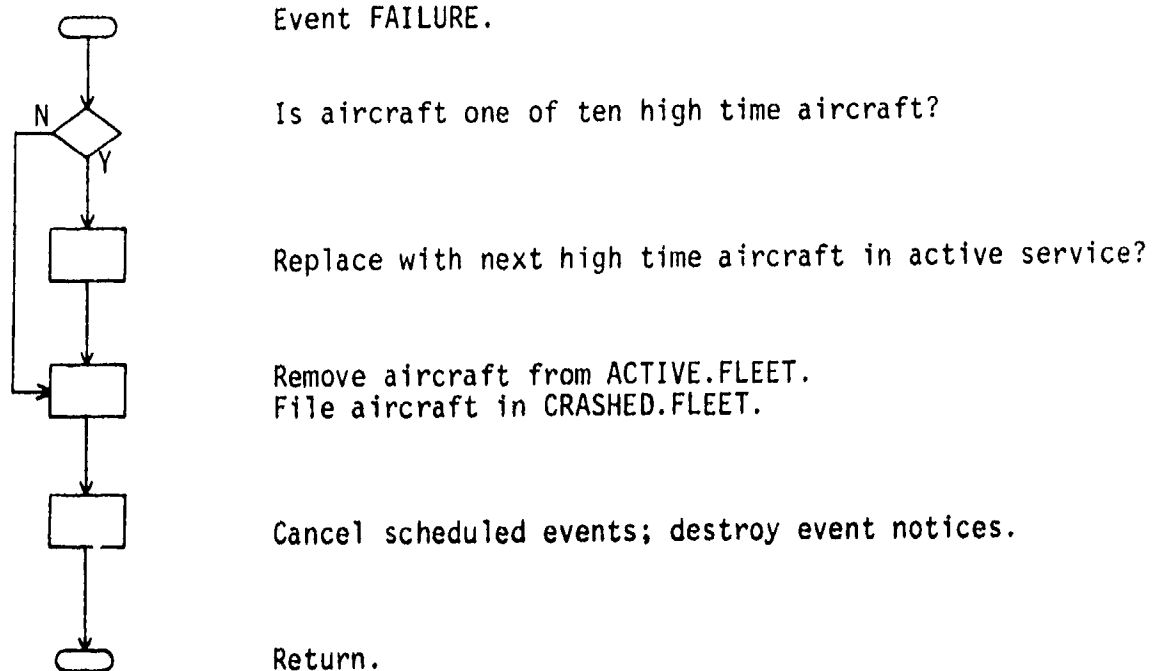
This event represents structural failure. When this event occurs, the aircraft is removed from the active fleet. If this aircraft was one of the ten high-time aircraft being monitored for the purpose of increasing inspection intervals, it is replaced by the next high-time aircraft in active service. Any remaining scheduled events are cancelled and their event notices destroyed. This event can be scheduled in events 1.STRENGTH.REDUCTION, 2.STRENGTH.REDUCTION, and 3.STRENGTH.REDUCTION.

#### 30.2 Local Variables

HOLD - This integer variable serves as an intermediate storage for aircraft identification numbers when replacing one of the ten high-time aircraft.

IDFA - This integer variable is the identification number of the aircraft under consideration.

#### 30.3 Flow Chart



### 31. RETIRE.FROM.SERVICE

#### 31.1 Description

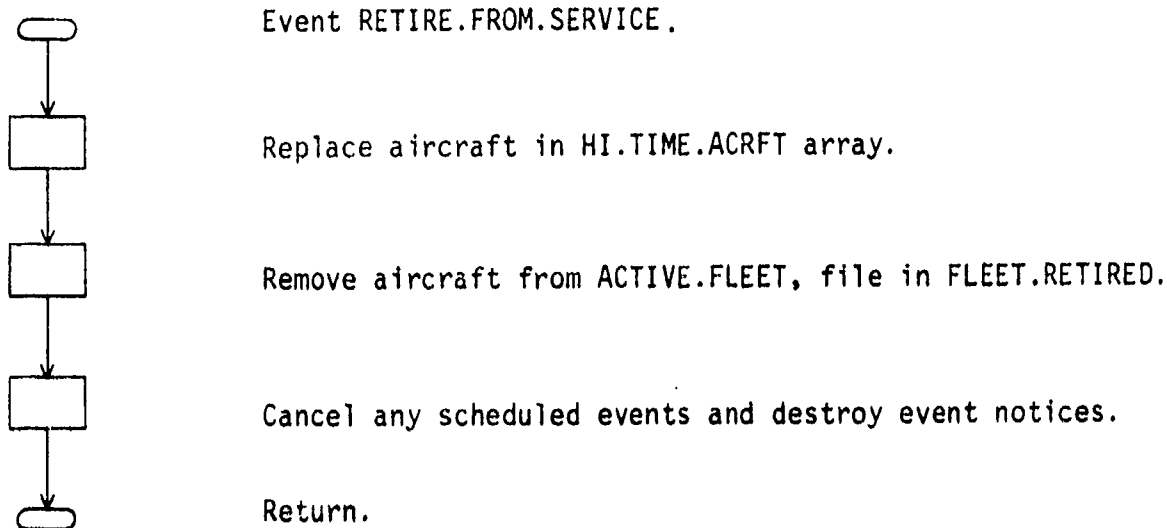
This event represents the retirement of an aircraft from active service. The aircraft is replaced in the HI.TIME.ACRFT array by that active aircraft not in the array which has the highest time in service. The aircraft being retired is removed from the set ACTIVE.FLEET and filed in the set FLEET.RETIRED. All remaining scheduled events for this aircraft are cancelled and the event notices destroyed. This event can only be scheduled in the event ENTER.SERVICE.

#### 31.2 Local Variables

HOLD - This integer variable is used as an intermediate storage to hold aircraft identification numbers during the HI.TIME.ACRFT replacement.

IDRET - This integer variable is the identification number of the aircraft being retired from service.

#### 31.3 Flow Chart



## 32. REPAIR

### 32.1 Description

This event represents the structural repair of an element. If there is a modification pending on the element, it is installed at this time. If the events FAILURE and REACH.FAILSAFE.LGT are scheduled, they are cancelled. It is assumed that all existing defects are repaired and that new times to defect occurrences are determined in the same manner as when the aircraft entered service. The size and number of all existing cracks are compared with the inspection interval decrease criteria. If these criteria are met, then the events which decrease inspection intervals and perform special fleet-wide inspections are scheduled. All defects that were scheduled but had not occurred by this time are not affected by this event and will occur as originally scheduled. This event can be scheduled in events D.LEVEL.INSPECTION, EXAMINE, and IMMEDIATE.FLEET.INSPECTION.

### 32.2 Local Variables

AAFL - This real variable is the actual average fatigue life of the element design.

CL - This real variable is the calculated fatigue crack length.

HOURS.TC.CORROSION - This real variable is the time in flight hours to corrosion initiation.

MAX.CRK - This real variable is the maximum crack length in the element.

RST - This real variable is the remaining service time of the aircraft.

STR.RED - This real variable is the element strength reduction because of all existing cracks.

TA1 - This real variable is the simulation time of the first crack initiation.

TA3 - This real variable is the simulation time of the third crack initiation.

CCL - This real variable is the element critical crack length.

FIRST.LIFE - This real variable is the time in flight hours to first crack initiation.

IDREP - This integer variable is the aircraft identification number.

POT.CRK - This real variable is the sum of the maximum crack length and the length that the crack will grow during the shortest internal inspection interval.

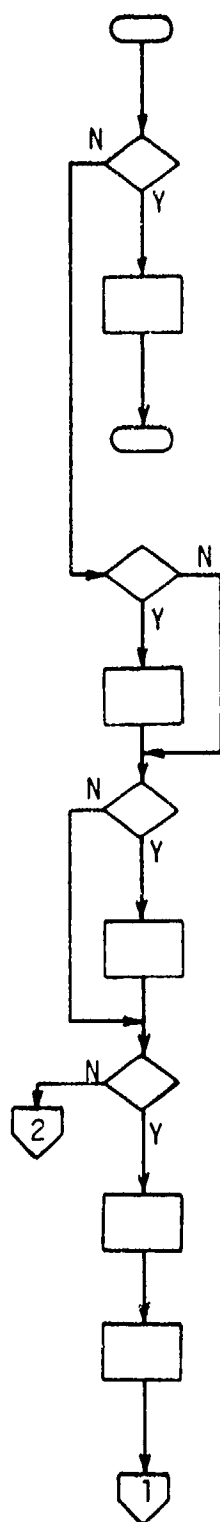
SECOND.LIFE - This real variable is the time in flight hours to the second crack initiation.

TAC - This real variable is the simulation time of the corrosion initiation.

TA2 - This real variable is the simulation time of the second crack initiation.

THIRD.LIFE - This real variable is the time in flight hours to the third crack initiation.

### 32.3 Flow Chart



Event REPAIR.

Is modification pending on this aircraft?

Install modification.

Return.

Is event FAILURE scheduled?

Cancel event FAILURE.

Is event REACH.FAIL.SAFE.LGT scheduled?

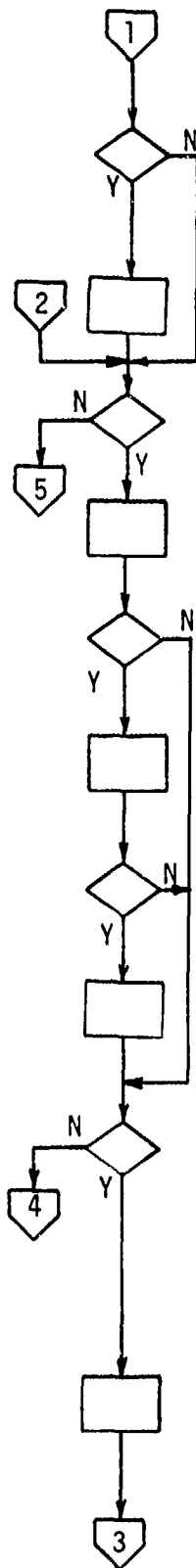
Cancel event REACH.FAIL.SAFE.LGT.

Does corrosion exist?

Set corrosion factor equal to 1.0.

Determine new time to corrosion initiation.





Is time to corrosion initiation less than remaining service life of aircraft?

Schedule corrosion initiation.

Does first crack exist?

Calculate crack length.

Does second crack exist?

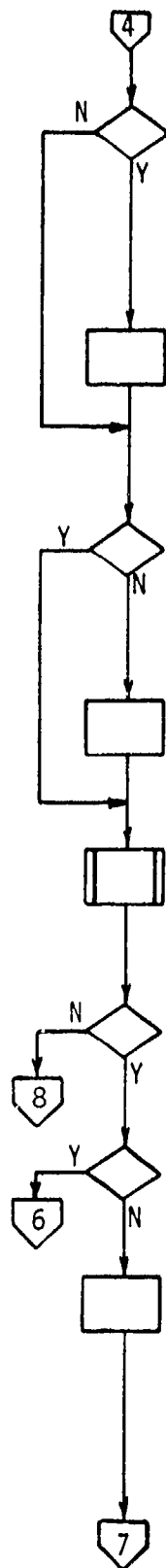
Calculate crack length.

Does third crack exist?

Calculate crack length.

Is sum of crack lengths plus crack growth rate times inspection interval greater than one-half the fail-safe crack length?

Increase inspection frequency and perform a special fleet wide inspection.



Is sum of crack lengths found in entire fleet greater than one-fifth of fleet size times fail-safe crack length?

Increase inspection frequency and perform a special fleet wide inspection.

Is there a modification pending somewhere in the fleet?

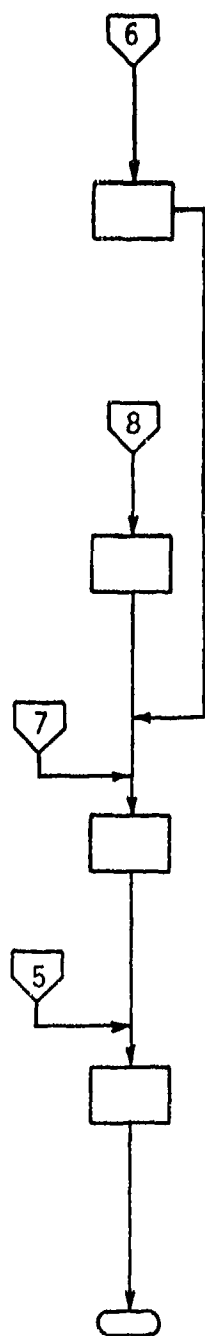
Schedule event DECISION.ON.MOD.

Call routine FATIGUE.LIFE.SCATTER.

Does second crack exist?

Does third crack exist?

Let time to first crack initiation be set to time to third crack initiation and times to second and third crack initiations be set to shortest times from routine FATIGUE.LIFE.SCATTER.



Let times to all three crack initiations be taken from FATIGUE.LIFE.SCATTER.

Set time to first crack to time to second crack. Set time to second crack to time to third crack. Set time to third crack from FATIGUE.LIFE.SCATTER.

Reschedule crack initiations for those cracks with times to crack initiation less than remaining service life of aircraft.

Cancel scheduled inspections below overhaul level.

Return.

### 33. T. INSPECTION: INCREASE

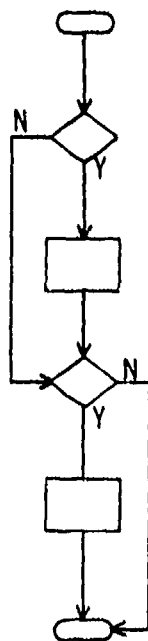
#### 33.1 Description

This event represents an inspection frequency increase for a particular aircraft pending a structural modification because of a fatigue test failure. The factor by which the inspection frequency for both the close internal and the close external inspection is increased depends on the element being considered and is an input parameter. This event can only be scheduled in event T.IMPLEMENT.MOD.

#### 33.2 Local Variables

IDTI - This integer variable is the identification number of the aircraft under consideration.

#### 33.3 Flow Chart



Event T.INSPECTION.INCREASE.

Is lowest internal level or lowest external level of inspection equal to C-level?

Increase C-level inspection frequency.

Is lowest internal level or lowest external level of inspection equal to D-level?

Increase D-level inspection frequency.

Return.

### 34. INCREASE.INSPECTION.FREQUENCY

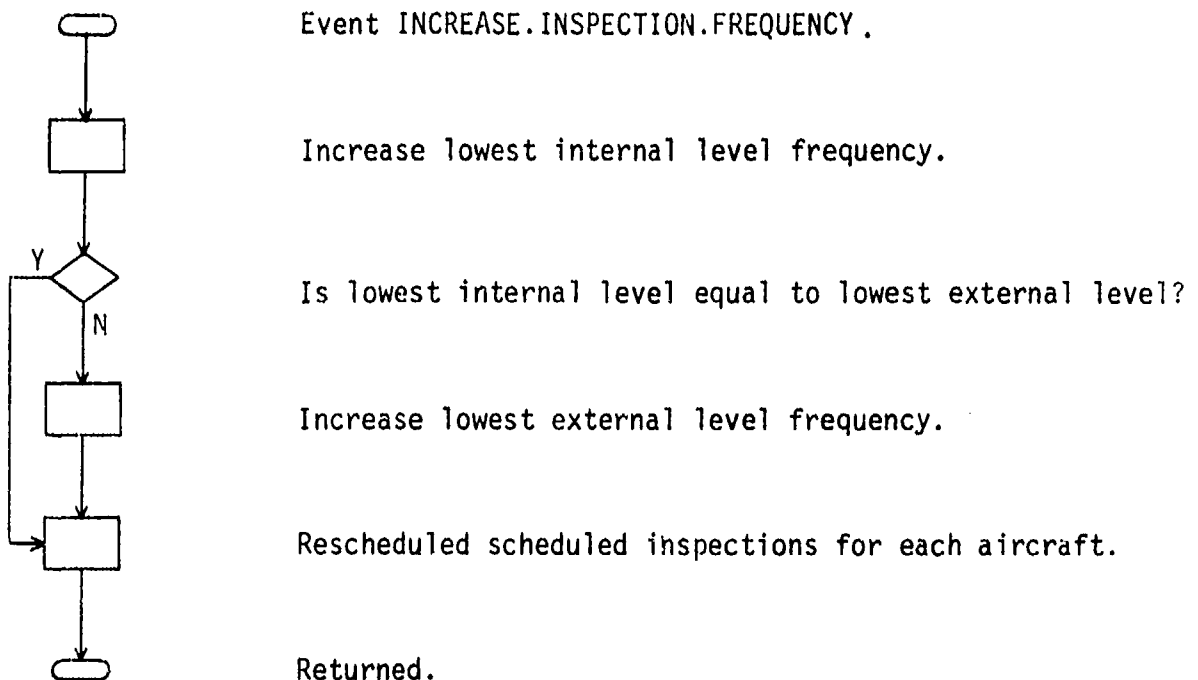
#### 34.1 Description

This event represents a fleet-wide increase in the frequencies of the lowest level internal and external inspections. These may or may not be the same levels. However, the A-level and B-level frequencies are never changed. All scheduled inspections are cancelled and rescheduled to reflect the frequency increase. This event can be scheduled in events REPAIR and IMMEDIATE.FLEET.INSPECTION.

#### 34.2 Local Variables

DIFF - This real variable is the difference in flight hours between the old and new lowest internal level inspection intervals.

#### 34.3 Flow Chart



### 35. IMMEDIATE.FLEET.INSPECTION

#### 35.1 Description

This event represents an immediate fleet-wide inspection caused by finding a defect considered too hazardous to depend on scheduled inspections for detection of additional defects. Existing crack lengths and corrosion areas are calculated along with the associated probabilities of detection. As in the scheduled inspections, these probabilities are compared with a random number to determine whether or not the defect is detected. This event is always preceded by the event INCREASE.INSPECTION.FREQUENCY. Defects found during this inspection can cause an additional increase in the frequency of normally scheduled inspections. This event can only be scheduled in event REPAIR.

#### 35.2 Local Variables

AREA - This real variable is the area in sq. inches of an existing corrosion defect.

CL - This real variable is the length in inches of an existing fatigue crack.

M1 - This real variable is the slow crack growth rate for a particular aircraft.

POT.CRK - This real variable is the sum of the maximum crack length detected on a particular aircraft plus the product of the average crack growth rate and the current lowest internal level inspection interval.

SR - This real variable is the sum of the lengths of all cracks detected on a particular aircraft.

TAC - This real variable is the time of initiation of an existing corrosion defect.

TA2 - This real variable is the time of initiation of an existing second fatigue crack.

CCL - This real variable is the critical crack length of the element under consideration.

FOUND - This real variable serves as a switch which is set equal to two when corrosion is detected and set equal to one when a crack is detected.

MAX.CRK - This real variable is the maximum crack length detected on a particular aircraft.

M2 - This real variable is the fast crack growth rate for a particular aircraft.

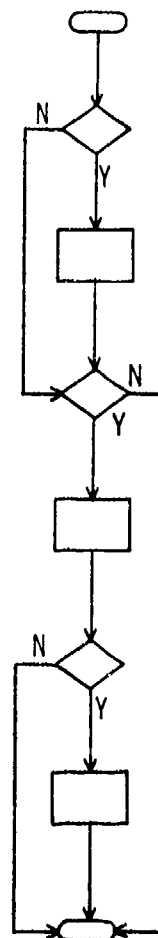
POT.STR.RED - This real variable is the sum of the detected crack lengths on a particular aircraft plus the product of the crack growth rate and the current lowest internal level inspection interval.

STR.RED - This real variable is equal to SR.

TA1 - This real variable is the time of initiation of an existing first fatigue crack.

TA3 - This real variable is the time of initiation of an existing third fatigue crack.

### 35.3 Flow Chart



Event IMMEDIATE.FLEET.INSPECTION.

Does corrosion exist?

Calculate corrosion area and test for detection.

Do cracks exist?

Calculate crack lengths and test for detection.

Do detected cracks meet criteria for inspection frequency increase?

Schedule event INCREASE.INSPECTION.FREQUENCY to occur immediately.

Return.

### 36. DECISION.ON.MOD

#### 36.1 Description

This event makes the decision on whether or not to develop a structural modification because of service experience. The decision to develop a modification is made by comparing the cost per flight hour of the modification with the repair cost per flight hour plus the increased inspection cost per flight hour. The modification cost per flight hour is found by dividing the total fleet modification cost by the remaining service life of the fleet. The repair cost per flight hour is found by dividing the total fleet repair costs since the last modification by the fleet flight time since the last modification. The increased inspection cost per flight hour is found by dividing the projected increased inspection costs by the remaining service life of the fleet. A modification is justified when

$$MCPH < RCPH + ICPH$$

where      MCPH = modification cost per flight hour  
             RCPH = repair cost per flight hour  
             ICPH = increased inspection cost per flight hour

This event can be scheduled only in event REPAIR.

#### 36.2 Local Variables

ACCUMULATED.HRS - This real variable is the total fleet time since the last modification.

MD.COST - This real variable is the cost of installing a modification on a single aircraft. The costs for additional modifications can differ from those for the initial modifications.

NFTS - This real variable is the total production time of all aircraft entering service after the second production rate goes into affect.

POST.MOD.HRS - This real variable is the total fleet service time remaining after the modification.

TOOLING - This real variable is the tooling cost in the development of a modification. The costs for additional modifications can differ from those for the initial modifications.

ICPH - This real variable is the increased inspection cost per flight hour.



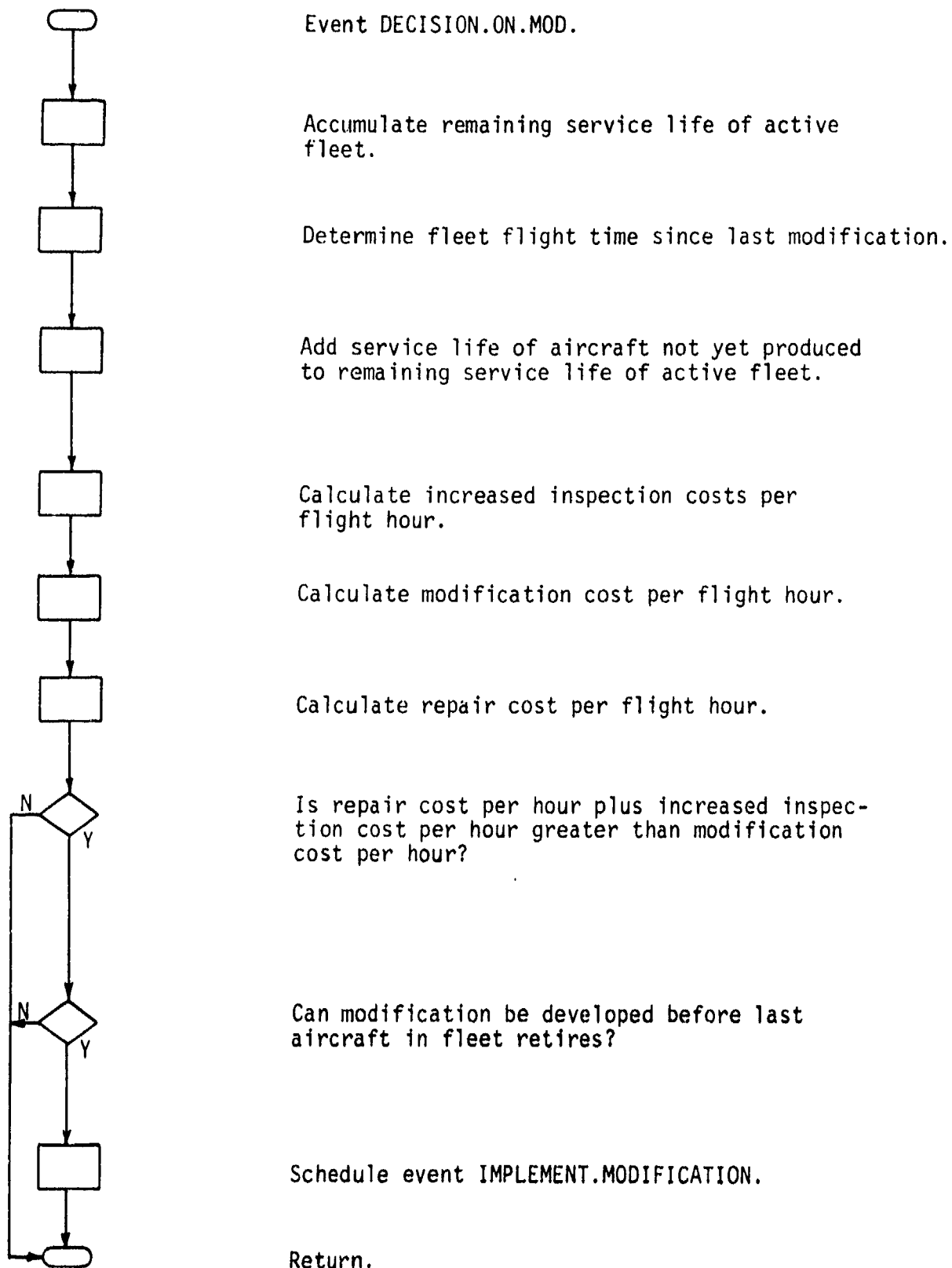
MCPH - This real variable is the modification cost per flight hour.

MRFH - This real variable is the service time remaining on a particular aircraft after its modification.

NPDL - This real variable is the number of aircraft which have entered service.

RCPH - This real variable is the repair cost per flight hour of the fleet.

### 36.3 Flow Chart



### 37. IMPLEMENT.MODIFICATION

#### 37.1 Description

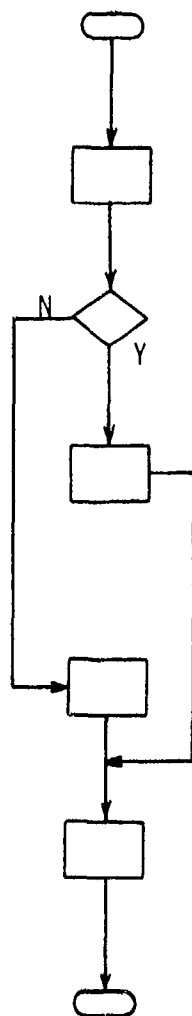
This event represents the development of a modification because of service experience. If the modification is to be fatigue tested, the actual average fatigue life is set equal to the original predicted life of the element design. Otherwise, the actual average fatigue life is determined by calling event REAL.LIFE. Elements of the alpha array SMOD.PENDING are set equal to "YES" to indicate which aircraft have service modifications pending. These modifications will be installed at the next D-level inspection or defect repair. Intervals for all levels of inspection are set to their initial values. This event can be scheduled only in event DECISION.ON.MOD.

#### 37.2 Local Variables

NSIG - This real variable is the standard deviation of the ratio distribution passed to routine REAL.LIFE.

NMU - This real variable is the mean of the ratio distribution passed to routine REAL.LIFE.

### 37.3 Flow Chart



Event IMPLEMENT.MODIFICATION.

Set inspection intervals to initial values.

Is modification tested?

Set actual average fatigue life equal to predicted average fatigue life.

Call routine REAL.LIFE.

Set alpha array SMOD.PENDING.

Return.

## 38. DISPLAY.OUTPUT

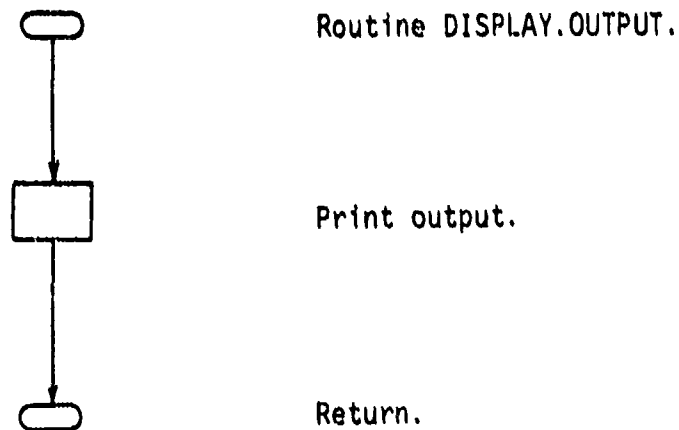
### 38.1 Description

This routine prints the standard output for each element. It is called from the MAIN program immediately after the completion of each element simulation. This output is suppressed if the long list option is in effect.

### 38.2 Local Variables

All the local variables are used to temporarily hold output values.

### 38.3 Flow Chart



### 39. SUMMARY

#### 39.1 Description

This routine prints the standard output summary for each element type. The output of routine DISPLAY.OUTPUT for all the elements in an element type is contained in this output summary. This routine is called from the MAIN program each time a new element type is read in and at the end of the program run. This output is suppressed if the long list option is in effect.

#### 39.2 Local Variables

All the local variables are used to temporarily hold output values.

#### 39.3 Flow Chart



Routine SUMMARY.

Print output.

Return.

APPENDIX B  
INITIAL PROGRAM SOURCE LISTING

THIS PAGE IS BEST QUALITY PRACTICALLY  
FROM COPY FURNISHED TO DDO

[illegible]



LINE CAGE SIMSCRIPT II.5 RELEASE RF 09/01/76 PAGE 2

```

54 **
55 ** INTEGER ARRAYS
56 **
57 DEFINE AISCN, A2SN, A3SN, AF, AIRPLANE, AAL, AHL, ACL, ADL, AC, AYII, ACIO,
58 AICN, AICN, SACIO, GICCN, GICCN, MICTF, ACRFT, APIU, SAPIU, STIM,
59 SSIIA, FLTH, SFLTH, ARESL, AQAC, AIP, AZE, ARE
60 AS INTEGER, 1-DIMENSIONAL ARRAYS
61 DEFINE TLID AS AN INTEGER, 2-DIMENSIONAL ARRAY
62 **
63 ** ALPHA ARRAYS
64 **
65 DEFINE WOFEL, FLPMET, 1-CH-EXISTS, 2-CH-EXISTS, 3-CH-EXISTS, CO-EXISTS,
66 SP-SCN, SSYAN, SFLW, AIL, FSM, IEL, IEP, IES,
67 TON-DEFINING, SNO-PENDING, INSP-SCN, 1-INT, 2-INT, 3-INT, C-INT
68 AS ALPHA, 1-DIMENSIONAL ARRAYS
69 **
70 DEFINE ELIN AS AN ALPHA, 2-DIMENSIONAL ARRAY
71 **
72 TEMPORARY ENTITIES
73 **
74 EVERY AIRCRAFT HAS AN ENTRY-TIME, A TAIL-ID
75 AND MAY BELONG TO AN ACTIVE-FLIGHT, A FLIGHT-RETIRED AND A CRASHED-FLIGHT
76 DEFINE TAIL-ID AS AN INTEGER VARIABLE
77 **
78 TALLY MICRO AS THE MINIMUM, MICRO AS THE MAXIMUM, MICRO AS THE MEAN OF MICRO
79 TALLY MICRO AS THE MINIMUM, MICRO AS THE MAXIMUM, MICRO AS THE MEAN OF MICRO
80 TALLY MICRO AS THE MINIMUM, MICRO AS THE MAXIMUM, MICRO AS THE MEAN OF MICRO
81 TALLY MICRO AS THE MINIMUM, MICRO AS THE MAXIMUM, MICRO AS THE MEAN OF MICRO
82 TALLY MICRO AS THE MINIMUM, MICRO AS THE MAXIMUM, MICRO AS THE MEAN OF MICRO
83 TALLY MICRO AS THE MINIMUM, MICRO AS THE MAXIMUM, MICRO AS THE MEAN OF MICRO
84 TALLY MICRO AS THE MINIMUM, MICRO AS THE MAXIMUM, MICRO AS THE MEAN OF MICRO
85 TALLY MICRO AS THE MINIMUM, MICRO AS THE MAXIMUM, MICRO AS THE MEAN OF MICRO
86 TALLY MICRO AS THE MINIMUM, MICRO AS THE MAXIMUM, MICRO AS THE MEAN OF MICRO
87 TALLY MICRO AS THE MINIMUM, MICRO AS THE MAXIMUM, MICRO AS THE MEAN OF MICRO
88 TALLY MICRO AS THE MINIMUM, MICRO AS THE MAXIMUM, MICRO AS THE MEAN OF MICRO
89 TALLY MICRO AS THE MINIMUM, MICRO AS THE MAXIMUM, MICRO AS THE MEAN OF MICRO
90 TALLY MICRO AS THE MINIMUM, MICRO AS THE MAXIMUM, MICRO AS THE MEAN OF MICRO
91 TALLY MICRO AS THE MINIMUM, MICRO AS THE MAXIMUM, MICRO AS THE MEAN OF MICRO
92 TALLY MICRO AS THE MINIMUM, MICRO AS THE MAXIMUM, MICRO AS THE MEAN OF MICRO
93 TALLY MICRO AS THE MINIMUM, MICRO AS THE MAXIMUM, MICRO AS THE MEAN OF MICRO
94 TALLY MICRO AS THE MINIMUM, MICRO AS THE MAXIMUM, MICRO AS THE MEAN OF MICRO
95 TALLY MICRO AS THE MINIMUM, MICRO AS THE MAXIMUM, MICRO AS THE MEAN OF MICRO
96 TALLY MICRO AS THE MINIMUM, MICRO AS THE MAXIMUM, MICRO AS THE MEAN OF MICRO
97 TALLY MICRO AS THE MINIMUM, MICRO AS THE MAXIMUM, MICRO AS THE MEAN OF MICRO
98 TALLY MICRO AS THE MINIMUM, MICRO AS THE MAXIMUM, MICRO AS THE MEAN OF MICRO
99 TALLY MICRO AS THE MINIMUM, MICRO AS THE MAXIMUM, MICRO AS THE MEAN OF MICRO
100 TALLY MICRO AS THE MINIMUM, MICRO AS THE MAXIMUM, MICRO AS THE MEAN OF MICRO
101 TALLY MICRO AS THE MINIMUM, MICRO AS THE MAXIMUM, MICRO AS THE MEAN OF MICRO
102 TALLY MICRO AS THE MINIMUM, MICRO AS THE MAXIMUM, MICRO AS THE MEAN OF MICRO
103 TALLY MICRO AS THE MINIMUM, MICRO AS THE MAXIMUM, MICRO AS THE MEAN OF MICRO
104 TALLY MICRO AS THE MINIMUM, MICRO AS THE MAXIMUM, MICRO AS THE MEAN OF MICRO
105 TALLY MICRO AS THE MINIMUM, MICRO AS THE MAXIMUM, MICRO AS THE MEAN OF MICRO
106 TALLY MICRO AS THE MINIMUM, MICRO AS THE MAXIMUM, MICRO AS THE MEAN OF MICRO

```

EVENT NOTICES INCLUDE ENTER-SERVICE, IMPLEMENTATION, IMPLEMENT-REIFICATION,

09/01/76 PAGE 3

LINE CACI SIMSCRIPT II.5 RELEASE AF

```

107  "
108  INCREASE.INSPECTION.FREQUENCY, IMMEDIATE.FLEET.INSPECTION,
109  DECISION.ON.WND
110  EVERY IN-SERVICE.DAMAGE HAS AN IDSDM
111  DEFINE IDSDM AS AN INTEGER VARIABLE
112  EVERY CONSUMPTION HAS AN IDCN
113  DEFINE IDCN AS AN INTEGER VARIABLE
114  EVERY 1-STRENGTH.REDUCTION HAS AN ID1SR
115  DEFINE ID1SR AS AN INTEGER VARIABLE
116  EVERY 2-STRENGTH.REDUCTION HAS AN ID2SR
117  DEFINE ID2SR AS AN INTEGER VARIABLE
118  EVERY 3-STRENGTH.REDUCTION HAS AN ID3SR
119  DEFINE ID3SR AS AN INTEGER VARIABLE
120  EVERY 1-ITE HAS AN ID1E
121  DEFINE ID1E AS AN INTEGER VARIABLE
122  EVERY 2-ITE HAS AN ID2E
123  DEFINE ID2E AS AN INTEGER VARIABLE
124  EVERY 3-ITE HAS AN ID3E
125  DEFINE ID3E AS AN INTEGER VARIABLE
126  EVERY A-LEVEL.INSPECTION HAS AN IDA
127  DEFINE IDA AS AN INTEGER VARIABLE
128  EVERY M-LEVEL.INSPECTION HAS AN IDM
129  DEFINE IDM AS AN INTEGER VARIABLE
130  EVERY C-LEVEL.INSPECTION HAS AN IDC
131  DEFINE IDC AS AN INTEGER VARIABLE
132  EVERY D-LEVEL.INSPECTION HAS AN IDD
133  DEFINE IDD AS AN INTEGER VARIABLE
134  EVERY I-INSPECTION.INCREASE HAS AN IIDI
135  DEFINE IIDI AS AN INTEGER VARIABLE
136  EVERY REPAIR HAS AN IDREP
137  DEFINE IDREP AS AN INTEGER VARIABLE
138  EVERY FAILURE HAS AN IDFA
139  DEFINE IDFA AS AN INTEGER VARIABLE
140  EVERY WEACH.FAIL.SAFF.LGT HAS AN IDWFL
141  DEFINE IDWFL AS AN INTEGER VARIABLE
142  EVERY RETIRE.FROM.SERVICE HAS AN IDRET
143  DEFINE IDRET AS AN INTEGER VARIABLE
144  "
145  PRIORITY ORDER IS REPAIR, INCREASE.INSPECTION.FREQUENCY,
146  IMMEDIATE.FLEET.INSPECTION, D.LEVEL.INSPECTION, C.LEVEL.INSPECTION,
147  M.LEVEL.INSPECTION
148  "
149  OFFICE POOD AS A REAL FUNCTION
150  DEFINE RATE AS A REAL FUNCTION
151  "
152  END

```

09/01/76 PAGE 4

```

LINE  CACT SIMSCRIPT II.5  RELEASE AF
1  MAIN
2  **
3  ** SET RANDOM NUMBER SEEDS
4  **
5  LET SEED.V(5) = SEED.V(4)
6  LET SEED.V(6) = SEED.V(5)
7  **
8  ** RESERVE AIRCRAFT ARRAYS
9  **
10 RESERVE HI.TIME.ACFT(*), WIND(*) AS 10
11 RESERVE SC(*), SD(*) AS 30
12 RESERVE ACID(*), FLTHR(*) AS 10
13 RESERVE APID(*), STIM(*) AS 20
14 RESERVE SACID(*), SFLTHR(*) AS 100
15 RESERVE SAPID(*), SSTIM(*) AS 200
16 RESERVE WDEL(*) AS 2 AND FLEMET(*) AS 4
17 RESERVE ANCH(*) AS 100, CARCD(*) AS 4
18 **
19 ** READ AIRCRAFT INPUT
20 **
21 READ MODEL(1), MODEL(2) AS 2 & 4
22 READ SIZE,UP,FLFET, USAGE,LIFE
23 READ BEGIN,PRODUCTION, TIME, 2.PRODUCTION,TIME, PNCMS
24 READ START,TEST, TEST.ACCEL,FAC, C.GRINTM,DATE
25 READ C7, C24, C20, MU,K, SIG,M, A, F
26 READ IANCH(1), IANCH(2), IANCH(3), IANCH(4)
27 READ CANCH(1), CARCD(2), CARCD(3), CARCD(4)
28 READ SAMPLING, LINC,LIST, PCCI
29 IF LONG,LIST = "YES"
30 READ NOF
31 RESERVE FLID(*,*) AS 4 BY NOF AND ANAC(*) AS NOF
32 RESERVE FLID(*,*) AS NINE BY *
33 FOR I = 1 TO NOF
34 DO
35 START NEW CARD
36 READ FLID(1,1), FLID(2,1), FLID(3,1), FLID(4,1) AS 4 & 4
37 READ ANAC(1)
38 RESERVE FLID(1,*) AS ANAC(1)
39 ALSO FOR J = 1 TO ANAC(1)
40 DO
41 READ FLID(1,1)
42 LOOP
43 ALWAYS
44 ** RESERVE FLEMET ARRAYS
45 **
46 RESERVE NICH(*), NICH(*), NICH(*), NICH(*), NICH(*) AS 4
47 RESERVE APPLA,EC(*), TEND,PRVING(*), SHIP,PE,OTING(*) AS SIZE,UP,FLFET
48 RESERVE C.INTERVAL(*), AIL(*), APSC(*), ESP(*) AS SIZE,UP,FLFET
49 RESERVE D.INTERVAL(*), OCCUR,ON(*), CREFP,TIME(*) AS SIZE,UP,FLFET
50 RESERVE COMF,TIME(*), LCP,EXISTS(*), PCW,EXISTS(*) AS SIZE,UP,FLFET
51 RESERVE 3.CP,EXISTS(*), CP,EXISTS(*), INSP,SC(*) AS SIZE,UP,FLFET
52 RESERVE AIS(*), ASP(*) AS 4, AISC(*) AS 4, AISC(*) AS 4, AISC(*) AS 4
53 RESERVE AIS(*) AS 4, AISC(*) AS 4, AISC(*) AS 4, AISC(*) AS 4

```

LINE CACI SYMSCRIPT IF.S RELEASE AF 09/01/76 PAGE 5

```

54 RESERVE AAL(0), AAL(0), AAL(0), AAL(0) AS SIZE OF FLEET
55 RESERVE I.INT(0), I.INT(0), I.INT(0), I.INT(0) AS SIZE OF FLEET
56 RESERVE WSR(0), WSR(0), LAST,SD(0) AS SIZE OF FLEET
57 RESERVE WRI(0), WRI(0) AS SIZE OF FLEET
58 RESERVE IFI(0), IFI(0), IFI(0) AS SIZE OF FLEET
59 RESERVE AIE, AIE, AIE AS SIZE OF FLEET
60 ..
61 ..
62 ..
63 'READ' ADD 1 TO COUNT.ELEMENT
64 START NEW CASE
65 READ ELEMENT(1), ELEMENT(2), ELEMENT(3), ELEMENT(4) AS A A
66 IF ELEMENT(1) = "END"
67 GO TO END
68 OTHERWISE
69 READ PREDICTED.LIFE, ACTUAL.AVG.FAT.LIFE, W1.WEAR, W2.WEAR
70 READ LMT.LIFE, LMT.LIFE, CMT.LIFE, FSAF.LIFE, WINT.DEFECT, WINT.DEFECT
71 READ CMT, SMT, WINT.DEFECT, WINT.DEFECT, LEAD.TIME, LEAD.TIME, S.FACT, CMT
72 READ FSAF.DEPREC, FSAF.DEPREC, FSAF.DEPREC, FSAF.DEPREC, FSAF.DEPREC, FSAF.DEPREC
73 READ A.DEPREC.COST, A.DEPREC.COST, C.DEPREC.COST, D.DEPREC.COST
74 READ WINT.DEPREC, WINT.DEPREC, WINT.DEPREC, WINT.DEPREC, WINT.DEPREC, WINT.DEPREC
75 READ S.DEPREC.COST, S.DEPREC.COST, S.DEPREC.COST, S.DEPREC.COST, S.DEPREC.COST, S.DEPREC.COST
76 READ L.CMT, L.CMT, L.CMT, L.CMT, L.CMT, L.CMT
77 ..
78 ..
79 ..
80 ..
81 ..
82 ..
83 ..
84 ..
85 ..
86 ..
87 ..
88 ..
89 ..
90 ..
91 ..
92 ..
93 ..
94 ..
95 ..
96 ..
97 ..
98 ..
99 ..
100 ..
101 ..
102 ..
103 ..
104 ..
105 ..
106 ..

```

20/01/76 PAGE 0

LINE CACT SIMSCRIPT II.5 REFLESE AF

```

107 REGARDLESS
108 LET IAPFL = ACTUAL.AVG.FAT.LIFE
109 LET APTS = (SIZE_OF_FLEET + 1 - TRUNC_F(PWMC/PRODUCTION.TIME)) *
110 2.PRODUCTION.TIME
111 LET SAIL = START.TIME + IAPFL/IFST.ACC.FACT * LEAD.TIME
112 IF SAIL LT NEXT.PRODUCTION + PWMC * HTS * MSGE.LIFE
113 IF PREDICTED.LIFE LT 2.0 * USAGE.LIFE AND IAPFL LT PREDICTED.LIFE OR
114 PREDICTED.LIFE GE 2.0 * USAGE.LIFE AND IAPFL LT 2.0 * MSGE.LIFE
115 SCHEDULE A 1 TO ELEMENT WHEN AT SAIL
116 LET TFS.FAILURE = "YES"
117 REGARDLESS
118 REGARDLESS
119 IF TIME = "YES"
120 START NEW PAGE
121 WRITE MODEL(1), "MODEL(1) AS S 50, "AIRCRAFT TYPE: ", P & Q
122 SKIP 2 OUTPUT LINE WITH TIME, "FLEET, USAGE, LIFE AS FOLLOWS
123 PRINT 1 DOUBLE LINE WITH TIME, "FLEET: "
124 SKIP 1 OUTPUT LINE
125 WRITE ELEMENT(1), ELEMENT(2), ELEMENT(3), ELEMENT(4)
126 AS S 50, "STRUCTURAL ELEMENT: ", A &
127 SKIP 2 OUTPUT LINES
128 IF PREDICTED.LIFE = 0.0
129 PRINT 1 DOUBLE LINE WITH IAPFL AS FOLLOWS
130 ELEMENT OF SIGN FATISUF TESTED WITH IAPFL IN PRODUCTION
131 JUMP AHEAD
132 ELSE
133 PRINT 1 DOUBLE LINE WITH PREDICTED.LIFE, IAPFL AS FOLLOWS
134 WITH
135 PRINT 1 LINE AS FOLLOWS
136
137 PRINT 1 OUTPUT LINE
138 PRINT 1 LINES WITH IAPFL(1), IAPFL(2), IAPFL(3), IAPFL(4) AS FOLLOWS
139
140
141
142
143
144
145
146
147
148
149
150
151

```

ACTUAL AVERAGE FATIGUE LIFE: 000000 MINUS

ACTUAL AVERAGE FATIGUE LIFE: 000000 MINUS

INITIAL INSPECTION INTERVALS

A-LEVEL 0000 MINUS

B-LEVEL 0000 MINUS

C-LEVEL 0000 MINUS

D-LEVEL 0000 MINUS

ALWAYS

SCHEDULE A 1 ENTER SERVICE AT NEXT PRODUCTION

START SIMULATION

IF LOGIC LIST = "YES"

CALL DISPLAY OUTPUT

ALWAYS

FOR EVERY AIRCRAFT IN CURRENT FLEET

DO

REMOVE AIRCRAFT (TAIL ID) FROM CHASER FLEET

DESTROY THE AIRCRAFT CALLED AIRCRAFT (TAIL ID)

LOOP

FOR EVERY AIRCRAFT IN FLEET, SET

DO

09/01/76 PAGE 7

```

LINE CACI SIMSCRIPT II.5 RELEASE AF
152 REMOVE AIRPLANE(TAIL.ID) FROM FLEET-RETIRE
153 DESTROY THE AIRCRAFT CALLED AIRPLANE(TAIL.ID)
154 LOOP
155 **
156 GO READ
157 'EOU' IF LONGLIST = 'NO'
158 CALL SUMMARY
159 ALWAYS
160 START NEW PAGE
161 PRINT LINE AS FOLLOWS
END OF SIMULATION
162 STOP
163 END

```

# LOCAL VARIABLES IN THIS ROUTINE

I.1	INTEGER	WORD 3	I.2	INTEGER	WORD 8
I.3	INTEGER	WORD 5	J.1	INTEGER	WORD 6
K.1	INTEGER	WORD 4	K.2	INTEGER	WORD 9
L.1	INTEGER	WORD 10	M.1	INTEGER	WORD 11
L.11	INTEGER	WORD 14	L.15	INTEGER	WORD 19
L.15	INTEGER	WORD 20	L.17	INTEGER	WORD 21
L.22	INTEGER	WORD 22	L.24	INTEGER	WORD 24
L.9	INTEGER	WORD 17	N.1	INTEGER	WORD 7
N.1	INTEGER	WORD 13	N.1	INTEGER	WORD 1
N.15	INTEGER	WORD 15			

09/01/76 PAGE 8

LINE CACI SIMSCRIPT II.5 RELEASE NF

```

1 ROUTINE INITIALIZATION
2
3 ** CHANGE INSPECTION LEVEL CODE TO NUMERIC VALUE
4
5 IF INT.LVL.INSPE = "A" LET ITRNL = 1 JUMP AHEAD
6 OTHERWISE
7 IF INT.LVL.INSPE = "H" LET ITRNL = 2 JUMP AHEAD
8 OTHERWISE
9 IF INT.LVL.INSPE = "C" LET ITRNL = 3 JUMP AHEAD
10 OTHERWISE
11 LET ITRNL = 4
12
13 **
14 IF EXT.LVL.INSPE = "A" LET EXT.INSPE.LEVEL = 1 JUMP AHEAD
15 OTHERWISE
16 IF EXT.LVL.INSPE = "H" LET EXT.INSPE.LEVEL = 2 JUMP AHEAD
17 OTHERWISE
18 IF EXT.LVL.INSPE = "C" LET EXT.INSPE.LEVEL = 3 JUMP AHEAD
19 OTHERWISE
20 LET EXT.INSPE.LEVEL = 4
21
22 **
23 ** CHANGE COMBUSTION RESISTANCE FACTOR TO MULTIPLYING FACTOR
24
25 IF CRF = 1 LET CRNF = 1.50 JUMP AHEAD
26 OTHERWISE
27 IF CRF = 2 LET CRNF = 1.25 JUMP AHEAD
28 OTHERWISE
29 IF CRF = 3 LET CRNF = 1.00 JUMP AHEAD
30 OTHERWISE
31 LET CRNF = 0.75
32
33 **
34 FOR I = 1 TO 4
35 LET AMCO(I) = 1/AMCO(I)
36 LET OICM(I) = 0
37 LET OICD(I) = 0
38
39 **
40 FOR I = 1 TO 10
41 LET MTIME-ACFT(I) = 1
42 LET WROD(I) = 0.0
43
44 **
45 LET FICK = 0
46 LET LMTA = 10
47 LET NICH = 1
48 LET NICH = 0
49 LET USOW = 0
50 LET USOW = 0
51 LET USOW = 0
52 LET USOW = 0
53 LET USOW = 0

```

09/01/76 PAGE 9

LINE CACI SINCRIPT II.5 RELEASE AF

```

54 LET MSFL = 0
55 LET NRFS = 0
56 LET WMD = 0
57 LET YSMD = 0
58 LET NSIC = 0
59 LET 1.NUM.OF.RFTIME = 0
60 LET 2.NUM.OF.CRASH = 0
61 LET ICA = 0
62 LET TIME.V = 0.0
63 LET AIRFRAME.TIME = 0.0
64 LET CUST.OF.REPAIRS = 0.0
65 LET FMSL = TARCH(1)
66 LET OTNSL = TARCH(2)
67 LET LIL = ITNM
68 LET LFL = EXT.INSPE.LEVEL
69 IF LFL = 1 OR LFL = 2
70 IF LIL = 1 OR LIL = 2
71 LET LIL = 3
72 ALWAYS
73 LET LEL = 3
74 REGARDLESS
75 LET T.FREQ.CMG = 1.0 - T.FREQ.CMG
76 LET S.FREQ.CMG = 1.0 - S.FREQ.CMG
77 LET PRE.OFCRASE = 1.0 + PRE.OFCRASE
78 LET LTM = "NO"
79 LET TES.FAILURE = "NO"
80 LET MOD.TESTED = "NO"
81 LET OFC.OF.WO.SCH = "NO"
82 LET IMP.SCH = "NO"
83 LET PREVIOUSLY.MODIFIED = "NO"
84
85 FOR I = 1 TO SIZE.OF.FLEFT
86 DO
87 LET IE1(I) = "NO"
88 LET IE2(I) = "NO"
89 LET IE3(I) = "NO"
90 LET CGRI(I) = 1.0
91 LET OCCUR.WO(I) = 0.0
92 LET AIL(I) = "NO"
93 LET FSM(I) = "NO"
94 LET INSP.SCH(I) = "NO"
95 LET TMM.PENDING(I) = "NO"
96 LET C.INTERVAL(I) = TARCH(3)
97 LET D.INTERVAL(I) = TARCH(4)
98 LET SWIN.PENDING(I) = "NO"
99 LET 1.CM.EXISTS(I) = "NO"
100 LET 2.CM.EXISTS(I) = "NO"
101 LET 3.CM.EXISTS(I) = "NO"
102 LET CM.EXISTS(I) = "NO"
103 LET SD.SCH(I) = "NO"
104 LET SD.SCH(I) = "NO"
105 LOOP
106
107 ** RESET TALLY COUNTERS

```



09/01/76 PAGE 10

LINE CACT SIMSCRIPT II.5 RELEASE AF

107     "     RESET TOTALS OF ICRT  
108     RESET TOTALS OF ICRT  
109     RESET TOTALS OF ISOT  
110     RESET TOTALS OF ACRKL  
111     RESET TOTALS OF MCRKL  
112     RESET TOTALS OF CCRKL  
113     RESET TOTALS OF DCRKL  
114     RESET TOTALS OF SCRKL  
115     RESET TOTALS OF ACA  
116     RESET TOTALS OF PCA  
117     RESET TOTALS OF CCA  
118     RESET TOTALS OF DCA  
119     RESET TOTALS OF SCA  
120     RETURN  
121     END  
122

THIS PAGE IS BEST QUALITY PRACTICABLE  
FROM COPY FURNISHED TO DDQ

09/01/76 PAGE 11

LINE CACI SINCE 11.5 RELEASE 46

1 RUNTIME SUM, INITIALIZE  
FOR I = 1 TO 4

2 DO

3 LET GCHC(1) = 0

4 LET GCHC(2) = 0

5 LET GCHC(3) = 0

6 LET GCHC(4) = 0

7 LET SASC = 0

8 LET SASC = 0

9 LET SASC = 0

10 LET SASC = 0

11 LET SASC = 0

12 LET GCHC = 0

13 LET GCHC = 0

14 LET GCHC = 0

15 LET GCHC = 0

16 LET GCHC = 0

17 LET GCHC = 0

18 LET GCHC = 0

19 RESET TOTALS OF GCHC

20 RESET TOTALS OF GCHC

21 RESET TOTALS OF GCHC

22 RESET TOTALS OF GCHC

23 RESET TOTALS OF GCHC

24 RESET TOTALS OF GCHC

25 RESET TOTALS OF GCHC

26 RESET TOTALS OF GCHC

27 RESET TOTALS OF GCHC

28 RESET TOTALS OF GCHC

29 RESET TOTALS OF GCHC

30 RESET TOTALS OF GCHC

31 RESET TOTALS OF GCHC

32 RESET TOTALS OF GCHC

33 RESET TOTALS OF GCHC

34 RESET TOTALS OF GCHC

35 RESET TOTALS OF GCHC

36 RESET TOTALS OF GCHC

00/21/76 PAGE 12

```

LINE CACI SCRIPT II.5 RELEASE 41
1 ROUTINE REAL.LIFE(MEAN,STD.DEV.,POL) YIELDING REL
2 " "
3 " " POL = PREDICTED AVERAGE FATIGUE LIFE OF ELEMENT REL
4 " " REL = ACTUAL AVERAGE FATIGUE LIFE OF ELEMENT DESIGN
5 " "
6 " " RATIO = RATIO OF ACTUAL LIFE / PREDICTED DESIGN
7 " " IF RATIO > 10, RATIO IS SET TO 10
8 " " IF RATIO < .1, RATIO IS SET TO .1
9 " " DISTRIBUTION OF RATIOS IS SET TO .1
10 " " MEAN = MEAN OF RATIOS IS REPRESENTED AS LOG NORMAL
11 " " STD.DEV = STANDARD DEVIATION OF RATIOS
12 " "
13 LET RATIO = LOG.NORMAL.F(MEAN,STD.DEV.,.1)
14 IF RATIO > 10.0
15 IF RATIO < 0.1
16 LET REL = RATIO * POL
17 RETURN
END

```

MEANLESS  
MEGAPRESS

LOCAL VARIABLES OF THIS ROUTINE

NAME	VALUE	TYPE	ADDRESS
RATIO	1	REAL	1000
REL	0	REAL	1004
POL	1	REAL	1008
MEAN	1	REAL	1012
STD.DEV	1	REAL	1016

09/01/76 PAGE 13

LINE CACI SIMSCRIPT II.5 RELEASE 8F

```

1  EVENT ENTER SERVICE
2  LET ILOC = ILOC + 1
3  LET ID = ILOC
4  CREATE AN AIRCRAFT CALLED AIRPLANE(ID)
5  LET TAIL.ID(AIRPLANE(ID)) = 10
6  LET ENTRY.TIME(AIRPLANE(ID)) = TIME.V
7  FILE AIRPLANE(ID) IN ACTIVE.FLEET
8  CALL FATIGUE.LIFE(SCATTER(ACTUAL.AVG.FAT.LIFE,A) YIELDING FIRST.LIFE,
9  SECOND.LIFE AND THIRD.LIFE
10 LET STD.FAST = M2.MEAN + .15
11 LET STD.SLOW = M1.MEAN + .15
12 LET MSR(ID) = RATE(M1.MEAN,STD.FAST,A)
13 LET MSR(ID) = RATE(M2.MEAN,STD.SLOW,A)
14 CALL PREDICT.CORROSION YIELDING HOURS.10.CORROSION
15 CALL PREDICT.SERVICE.DAMAGE YIELDING HOURS.TU.SERVICE.DAMAGE
16 IF HOURS.TU.CORROSION(ID) AT TIME.V + HOURS.10.CORROSION
17 SCHEDULE A CORROSION(ID) AT TIME.V + HOURS.10.CORROSION
18 LET AC(ID) = CORROSION
19 LET COMEP.TIME(ID) = TIME.V
20 LET CO.EXISTS(ID) = "MS"
21 REGARDLESS
22 **
23 IF HOURS.TU.SERVICE.DAMAGE LT USAGE.LIFE
24 SCHEDULE AN TU.SERVICE.DAMAGE(ID) AT TIME.V + HOURS.TU.SERVICE.DAMAGE
25 LET LAST.SD(ID) = TIME.V
26 LET SD.SCH(ID) = "YES"
27 REGARDLESS
28 IF RANDOM.F(7) LE MIN.TM.OBJECT.PROBABILITY
29 LET AN = RANDOM.F(3)
30 LET DEFECT.LIFE = (-4042.6 + LOG.E.F(MR))**.0132
31 ADD 1 TO GPO
32 IF DEFECT.LIFE LT FIRST.LIFE
33 LET FIRST.LIFE = DEFECT.LIFE
34 REGARDLESS
35 REGARDLESS
36 IF LTWO = "YES"
37 FOR I = 1 TO MAX(LDZ)
38 DO
39 IF ID = TLID(LDZ,I)
40 LET LIST = 1.0
41 SKIP 1 INPUT LINE
42 PRINT 1 LINE WITH ID, TIME.V AS FOLLOWS
43 A/C NO. *** ENTERS SERVICE ***** HOURS FROM START OF SIMULATION
44 IF DEFECT.LIFE GT 0.0
45 PRINT 1 LINE AS FOLLOWS
46 PRODUCTION DEFECT PRESENT
47 ALWAYS
48 SKIP 1 INPUT LINE
49 PRINT 3 LINES WITH FIRST.LIFE, SECOND.LIFE, THIRD.LIFE AS FOLLOWS
50 1ST CRACK INITIATION PROJECTED AT ***** FLIGHT HOURS
51 2ND CRACK INITIATION PROJECTED AT ***** FLIGHT HOURS
52 3RD CRACK INITIATION PROJECTED AT ***** FLIGHT HOURS

```

LINE CACT SIMSCRIPT II.5 RELEASE 8F 09/01/76 PAGE 12

```

40 PRINT 2 LINES WITH MSR(ID), MPR(ID) AS FOLLOWS
    SLOW CRACK GROWTH RATE = .000000 INCHES/MIN
    FAST CRACK GROWTH RATE = .000000 INCHES/HOUR
50 LEAVE
51 ELSE
52 LOOP
53 ALWAYS
54
55 IF FIRST.LIFE LT USAGE.LIFE OR CO.EXISTS(ID)="NS" OR SD.SCH(ID)="YES"
56 SCHEDULE A 1.STRENGTH.REDUCTION(ID) AT TIME.V + FIRST.LIFE
57 LET AISR(ID) = 1.STRENGTH.REDUCTION
58 LET CREP.TIME(ID) = TIME.V
59 LET 1.CR.EXISTS(ID) = "NS"
60 IF SECOND.LIFE LT USAGE.LIFE OR CO.EXISTS(ID)="NS" OR SD.SCH(ID)="YES"
61 SCHEDULE A 2.STRENGTH.REDUCTION(ID) AT TIME.V + SECOND.LIFE
62 LET ARSR(ID) = 2.STRENGTH.REDUCTION
63 LET 2.CR.EXISTS(ID) = "NS"
64 IF THIRD.LIFE LT USAGE.LIFE OR CO.EXISTS(ID)="NS" OR SD.SCH(ID)="YES"
65 SCHEDULE A 3.STRENGTH.REDUCTION(ID) AT TIME.V + THIRD.LIFE
66 LET AISR(ID) = 3.STRENGTH.REDUCTION
67 LET 3.CR.EXISTS(ID) = "NS"
68 REGARDLESS
69 REGARDLESS
70 REGARDLESS
71 SCHEDULE A 0.LEVEL.INSPECTION(ID) AT TIME.V + 0.INTERVAL(ID)
72 LET ADL(ID) = 0.LEVEL.INSPECTION
73 SCHEDULE A 4.TIME.PROV.SERVICE(ID) AT TIME.V + USAGE.LIFE
74 IF INCK EQ SIZE.OF.FLLET
75 RETURN
76 OTHERWISE
77 IF TIME.V = BEGIN.PRODUCTION OF PRCHG
78 SCHEDULE AN ENTER.SERVICE AT TIME.V + 2.PRODUCTION.TIME
79 RETURN
80 OTHERWISE
81 SCHEDULE AN ENTER.SERVICE AT TIME.V + PRODUCTION.TIME
82 RETURN
83 END

```

# LOCAL VARIABLES OF THIS ROUTINE

DEFECT.LIF	DOUBLE	WORD 17	FIRST.LIFE	DOUBLE	WORD 1
WORDS.T0.C	DOUBLE	WORD 11	1.1	INTEGER	WORD 25
I.2	INTEGER	WORD 26	1.3	INTEGER	WORD 27
J.1	INTEGER	WORD 28	K.1	INTEGER	WORD 30
K.2	INTEGER	WORD 31	K.3	INTEGER	WORD 32
K.4	INTEGER	WORD 33	L.4	INTEGER	WORD 31
L.6	INTEGER	WORD 34	LIST	DOUBLE	WORD 19
M.1	INTEGER	WORD 29	WORDS.T0.SP	DOUBLE	WORD 13
M.1	DOUBLE	WORD 23	AN	DOUBLE	WORD 15
SECOND.LIF	DOUBLE	WORD 3	STD.FAST	DOUBLE	WORD 7
STD.SLOW	DOUBLE	WORD 9	THIRD.LIFE	DOUBLE	WORD 5

09/01/76 PAGE 15

LINE CACI SIMSCRIPT II.5 RELEASE 8F

```

1  INITIATE FATIGUE.LIFE, SCATTER(MEL,N) YIELDING FIRST.LIFE, SECOND.LIFE AND
2  THIRD.LIFE
3  **
4  ** GENERATES FLEWAT FATIGUE LIVES REFLECTING BASIC FATIGUE SCATTER AND LOAD
5  ** ENVIRONMENT VARIATION
6  **
7  ** DISTRIBUTION OF FATIGUE LIVES IS REPRESENTED AS TRIU PARAMETER *FINULL
8  ** ALPHA = SHAPE PARAMETER
9  ** BETA = SCALE PARAMETER (CHARACTERISTIC VALUE)
10 ** N = RANDOM NUMBER STREAM
11 **
12  DEFINE N AS AN INTEGER VARIABLE
13  DEFINE LIFE AS A REAL, 1-DIMENSIONAL ARRAY
14  RESERVE LIFE(*) AS 3
15  LET ALPHA = 2.78
16  LET BETA = 4617.8001M
17  LET M4 = MAXIMUM.F.(2)
18  FOR I = 1 TO 3, DO
19    LET LIFE(I) = BETA * (LOG.E.(1/M4)) ** (1/ALPHA)
20    LET M4 = UNIFORM.F.(0.0,M4,1)
21  LOOP
22  LET FIRST.LIFE = LIFE(1)
23  LET SECOND.LIFE = LIFE(2)
24  LET THIRD.LIFE = LIFE(3)
25  RELEASE LIFE(*)
26  RETURN
27  END

```

LOCAL VARIABLES OF THIS ROUTINE

ALPHA	DOUBLE	WORD 13	BETA	DOUBLE	WORD 15
FIRST.LIFE	DOUBLE	WORD 5	LIFE	REAL	WORD 11
N	INTEGER	WORD 3	M4	DOUBLE	WORD 1
M4	DOUBLE	WORD 17	SECOND.LIFE	DOUBLE	WORD 7
THIRD.LIFE	DOUBLE	WORD 9			

09/01/76 PAGE 16

LINE CACI SIMSCRIPT II.5 RELEASE AF

```

1 ROUTINE INSTALL MODIFICATION
2
3 ** THIS ROUTINE REPRESENTS THE INSTALLATION OF A STRUCTURAL MODIFICATION CAUSED
4 BY A FATIGUE TEST FAILURE OR BY A SERVICE DEFECT FOUND IN THE ELEMENT
5
6 LET (SV) = VSND + 1
7 LET REEAMODIFIED = REEAMODIFIED + 1
8 LET C.INTERVAL(ID) = LARCD(3)
9 LET D.INTERVAL(ID) = LARCD(4)
10 LET RECURMOD(ID) = TIME.V
11
12 ** CANCEL SCHEDULED INSPECTIONS
13
14 IF INSP.SCH(ID) = "YES"
15 CALL CANCEL.SCHEDULED.INSPECTIONS
16 REGARDLESS
17
18 IF FSM(ID) = "YES"
19 LET FSM(ID) = "NO"
20 LET FAILURE = AF(ID)
21 CANCEL THE FAILURE
22 DESTROY THE FAILURE
23 ALWAYS
24 CALL FATIGUE.LIFE.SCATTER(ACTUAL.AVG.FAT.LIFE,A) YIELDING FIRST.LIFE.
25 SECOND.LIFE AND THIRD.LIFE
26 LET STD.SLOX = M1.MEAN + .15
27 LET STD.FAST = M2.MEAN + .15
28 LET WSR(ID) = RATE(M1.MEAN,STD.SLOX,S)
29 LET WFR(ID) = RATE(M2.MEAN,STD.FAST,S)
30 LET WST = USAGE.LIFE - TIME.V + FATIGUE.LIFE(1) * (1/WSR(ID) + 1/WFR(ID))
31
32 ** CANCEL PREVIOUSLY SCHEDULED CRACK AND CORROSION INITIATIONS, RESCHEDULE
33 THESE EVENTS IF THEY OCCUR WITHIN THE REMAINING SERVICE LIFE (RST) OF THE
34 AIRCRAFT
35
36 IF CO.EXISTS(ID) NE "NN"
37 LET CORROSION = AC(ID)
38 IF CO.EXISTS(ID) = "AS"
39 CANCEL THE CORROSION
40 ALWAYS
41 DESTROY THE CORROSION
42 LET CO.EXISTS(ID) = "NN"
43 LET CORN(ID) = 1.0
44 REGARDLESS
45 CALL PREDICT CORROSION YIELDING MUORS.ID,CORROSION
46 IF MUORS.ID,CORROSION LT WST
47 SCHEDULE A CORROSION AT TIME.V + MUORS.ID,CORROSION
48 LET AC(ID) = CORROSION
49 LET CORN.TIME(ID) = TIME.V
50 LET CO.EXISTS(ID) = "AS"
51 REGARDLESS
52
53 ** IF WANDOW.F(7) LE HIGHM.DEFECT.PRIORABILITY

```

THIS PAGE IS BEST QUALITY PRACTICABLE  
FROM COPY FURNISHED TO DDQ

09/01/76 PAGE 17

```

LINE  CACI SIMSCRIPT II.5  RELEASE AF
54      LET UPD = OPD + 1
55      LET RN = RANDOM.F(3)
56      LET DEFECT.LIFE = (-4042.6 * LOG.E.F(RN)) + 1.0132
57      IF DEFECT.LIFE LT FIRST.LIFE
58      LET FIRST.LIFE = DEFECT.LIFE
59      REGARDLESS
60      REGARDLESS
61      "
62      IF 1.CR.EXISTS(ID) NE "NN"
63      LET 1.STRENGTH.REDUCTION = AISR(ID)
64      IF 1.CR.EXISTS(ID) = "NS"
65      CANCEL THE 1.STRENGTH.REDUCTION
66      REGARDLESS
67      DESTROY THE 1.STRENGTH.REDUCTION
68      LET 1.CR.EXISTS(ID) = "NN"
69      IF 2.CR.EXISTS(ID) NE "NN"
70      LET 2.STRENGTH.REDUCTION = APSH(ID)
71      IF 2.CR.EXISTS(ID) = "NS"
72      CANCEL THE 2.STRENGTH.REDUCTION
73      REGARDLESS
74      DESTROY THE 2.STRENGTH.REDUCTION
75      LET 2.CR.EXISTS(ID) = "NN"
76      IF 3.CR.EXISTS(ID) NE "NN"
77      LET 3.STRENGTH.REDUCTION = ASSH(ID)
78      IF 3.CR.EXISTS(ID) = "NS"
79      CANCEL THE 3.STRENGTH.REDUCTION
80      REGARDLESS
81      DESTROY THE 3.STRENGTH.REDUCTION
82      LET 3.CR.EXISTS(ID) = "NN"
83      REGARDLESS
84      REGARDLESS
85      IF IE1(ID) = "YES"
86      LET 1.ITE = AIE(ID)
87      LET IE1(ID) = "NN"
88      CANCEL THE 1.ITE
89      DESTROY THE 1.ITE
90      IF IE2(ID) = "YES"
91      LET 2.ITE = APE(ID)
92      LET IE2(ID) = "NN"
93      CANCEL THE 2.ITE
94      DESTROY THE 2.ITE
95      IF IE3(ID) = "YES"
96      LET 3.ITE = AYE(ID)
97      LET IE3(ID) = "NN"
98      CANCEL THE 3.ITE
99      DESTROY THE 3.ITE
100     ALWAYS
101     ALWAYS
102     ALWAYS
103     ALWAYS
104     IF AIL(ID) = "YES"
105     LET REACH.FAIL.SAFE.LGT = AMFSL(ID)
106     CANCEL THE REACH.FAIL.SAFE.LGT

```



09/01/76 PAGE 1A

LINE CACI SIMSCRIPT II.5 RELEASE RF

```

107 DESTROY THE REACH.FAIL.SAFE.LGT
108 LET AIL(ID) = "NO"
109 REGARDLESS
110 ..
111 IF FIRST.LIFE LT RST OR CO.EXISTS(ID) = "NS" OR SD.SCH(ID) = "YES"
112 SCHEDULE A 1.STRENGTH.REDUCTION(ID) AT TIME.V + FIRST.LIFE
113 LET A1SK(ID) = 1.STRENGTH.REDUCTION
114 LET CRREP.TIME(ID) = TIME.V
115 LET 1.CR.EXISTS(ID) = "NS"
116 IF SECOND.LIFE LT RST OR CO.EXISTS(ID) = "NS" OR SD.SCH(ID) = "YES"
117 SCHEDULE A 2.STRENGTH.REDUCTION(ID) AT TIME.V + SECOND.LIFE
118 LET A2SK(ID) = 2.STRENGTH.REDUCTION
119 LET 2.CR.EXISTS(ID) = "NS"
120 IF THIRD.LIFE LT RST OR CO.EXISTS(ID) = "NS" OR SD.SCH(ID) = "YES"
121 SCHEDULE A 3.STRENGTH.REDUCTION(ID) AT TIME.V + THIRD.LIFE
122 LET A3SK(ID) = 3.STRENGTH.REDUCTION
123 LET 3.CR.EXISTS(ID) = "NS"
124 REGARDLESS
125 REGARDLESS
126 REGARDLESS
127 ..
128 IF LTIM = "YES"
129 FOR I = 1 TO NMAC(LDX)
130 DO
131 IF IO = TLID(LOX,I)
132 SKIP 1 OUTPUT LINE
133 PRINT 6 LINES WITH ID, TIME.V-ENTRY.TIME(AIRPLANE(ID)), FIRST.LIFE,
134 SECOND.LIFE, THIRD.LIFE, MSR(ID), MFR(ID) AS FOLLOWS
      MODIFICATION INSTALLED ON A/C NO. *** AT ***** FLIGHT HOURS
      1ST CRACK INITIATION PROJECTED AT ***** FLIGHT HOURS
      2ND CRACK INITIATION PROJECTED AT ***** FLIGHT HOURS
      3RD CRACK INITIATION PROJECTED AT ***** FLIGHT HOURS
      SLOW CRACK GROWTH RATE = ***** INCHES/HOUR
      FAST CRACK GROWTH RATE = ***** INCHES/HOUR
135 LEAVE
136 ELSE
137 LOOP
138 ALWAYS
139 RETURN
140 END

```

LOCAL VARIABLES OF THIS ROUTINE

OBJECT.LIF	DOUBLE	WORD 17	FIRST.LIFE	DOUBLE	WORD 1
MOURS.TO.C	DOUBLE	WORD 13	I.1	INTEGER	WORD 23
I.2	INTEGER	WORD 24	I.3	INTEGER	WORD 25
J.1	INTEGER	WORD 26	K.1	INTEGER	WORD 28
K.2	INTEGER	WORD 29	K.3	INTEGER	WORD 30
K.4	INTEGER	WORD 31	L.4	INTEGER	WORD 19
N.1	INTEGER	WORD 27	R.1	DOUBLE	WORD 21
RN	DOUBLE	WORD 15	RST	DOUBLE	WORD 11
SECOND.LIF	DOUBLE	WORD 3	STD.FAST	DOUBLE	WORD 9

THIS PAGE IS BEST QUALITY PRACTICABLE  
FROM COPY FURNISHED TO DDC

09/01/76 PAGE 19

LINE CACT SINSRIPT 11.5 RELEASE RF  
STD.SLOW DOUBLE WORD 7 THIND.LIFE DOUBLE WORD 5

09/01/76 PAGE 20

LINE CACT SYMSRIPT II.5 RELEASE RF

```

1 EVENT IN.SERVICE.DAMAGE(IDSNM)
2 **
3 ** REPRESENTS THE OCCURRENCE OF A SERVICE DAMAGE DEFECT; RESULTS IN IMMEDIATE
4 ** INITIATION OF NEXT SCHEDULED CRACK
5 **
6 **
7 **
8 **
9 **
10 **
11 **
12 **
13 **
14 **
15 **
16 **
17 **
18 **
19 **
20 **
21 **
22 **
23 **
24 **
25 **
26 **
27 **
28 **
29 **
30 **
31 **
32 **
33 **
34 **
35 **
36 **
37 **
38 **
39 **
40 **
41 **
42 **
43 **
44 **
45 **
46 **
47 **
48 **
49 **
50 **
51 **

```

DEFIN: IDSNM AS AN INTEGER VARIABLE  
 LET ID = IDSNM  
 IF ID = "YES"  
 FOR I = 1 TO MAXC(LDR)  
 ON  
 IF ID = ALLC(LDR, I)  
 LET LIST = 1.0  
 SET I INPUT LINE  
 PRINT 2 LINES WITH ID, TIME.V-ENTRY.TIME(AIRPLANE(ID)) AS FOLLOWS  
 A/C NO. \*\*\* EXPERIENCES SERVICE DAMAGE AT \*\*\*\*\* FLIGHT HOURS  
 CAUSES IMMEDIATE INITIATION OF NEXT SCHEDULED CRACK  
 LEAVE  
 FLSP  
 LOOP  
 ALWAYS  
 LET IDSNM = IDSNM + 1  
 LET GOSNM = GOSNM + 1  
 LET ISDT = TIME.V - LAST.SD(ID)  
 LET LISD = TIME.V - LAST.SD(ID)  
 CALL PREDICT.SERVICE.DAMAGE.YIELDING (GOSNM, ID, SERVICE.DAMAGE  
 LET PST = USAGE.LIFE - TIME.V + ENTRY.TIME(AIRPLANE(ID))  
 LET SO.SCH(ID) = "NO"  
 IF GOSNM, ID, SERVICE.DAMAGE (T.WST  
 SCHEDULE AN IN.SERVICE.DAMAGE(ID) AT TIME.V + (GOSNM, ID, SERVICE.DAMAGE  
 LET LAST.SD(ID) = TIME.V  
 LET SO.SCH(ID) = "YES"  
 REGARDLESS  
 \*\* TEST FOR SCHEDULED FIRST CRACK INITIATION  
 \*\*  
 LET 1.STRENGTH.REDUCTION = AIS\*(ID)  
 LET 2.STRENGTH.REDUCTION = APS\*(ID)  
 LET 3.STRENGTH.REDUCTION = AIS\*(ID)  
 IF 1.CR.EXISTS(ID) = "NS"  
 CANCEL THE 1.STRENGTH.REDUCTION  
 RESCHEDULE THE 1.STRENGTH.REDUCTION(ID) \*\*\*  
 IF SO.SCH(ID) = "NO" AND TIME.V > AIS\*(ID, STRENGTH.REDUCTION) OF WST + TIME.V  
 AND CO.EXISTS(ID) = "NS"  
 CANCEL THE 2.STRENGTH.REDUCTION  
 OFSTWY THE 2.STRENGTH.REDUCTION  
 CANCEL THE 3.STRENGTH.REDUCTION  
 DESTROY THE 3.STRENGTH.REDUCTION  
 LET 2.CR.EXISTS(ID) = "NO"  
 LET 3.CR.EXISTS(ID) = "NO"  
 ALWAYS  
 RETURN  
 OTHERWISE

09/01/76 PAGE 21

```

LINE  CACI SCRIPT 11.5  RELEASE RF
52  ** TEST FOR SCHEDULED SECOND CRACK INITIATION
53  **
54  IF 2.CM.EXISTS(ID) = "NS"
55  CANCEL THE 2.STRENGTH.REDUCTION
56  RESCHEDULE THE 2.STRENGTH.REDUCTION(ID) NOW
57  IF 2.SCH(ID) = "NO" AND TIME.ALS.STRENGTH.REDUCTION GE #51 + TIME.V
58  AND CO.EXISTS(ID) = "NS"
59  CANCEL THE 3.STRENGTH.REDUCTION
60  DESTROY THE 3.STRENGTH.REDUCTION
61  LET 3.CM.EXISTS(ID) = "N"
62  ALWAYS
63  RETURN
64  OTHERWISE
65  ** TEST FOR SCHEDULED THIRD CRACK INITIATION
66  **
67  **
68  IF 3.CM.EXISTS(ID) = "NS"
69  CANCEL THE 3.STRENGTH.REDUCTION
70  RESCHEDULE THE 3.STRENGTH.REDUCTION(ID) NOW
71  RETURN
72  OTHERWISE
73  RETURN
74  END

```

LOCAL VARIABLES IN THIS PROGRAM

1.1	INTEGER	MODE 9	1.2	INTEGER	MODE 10
1.3	INTEGER	MODE 11	1.3.1	INTEGER	MODE 11
2.1	INTEGER	MODE 12	2.1.1	INTEGER	MODE 12
2.2	INTEGER	MODE 13	2.2.1	INTEGER	MODE 13
2.3	INTEGER	MODE 14	2.3.1	INTEGER	MODE 14
2.4	INTEGER	MODE 15	2.4.1	INTEGER	MODE 15
2.5	INTEGER	MODE 16	2.5.1	INTEGER	MODE 16
2.6	INTEGER	MODE 17	2.6.1	INTEGER	MODE 17
2.7	INTEGER	MODE 18	2.7.1	INTEGER	MODE 18
2.8	INTEGER	MODE 19	2.8.1	INTEGER	MODE 19
2.9	INTEGER	MODE 20	2.9.1	INTEGER	MODE 20
2.10	INTEGER	MODE 21	2.10.1	INTEGER	MODE 21
2.11	INTEGER	MODE 22	2.11.1	INTEGER	MODE 22
2.12	INTEGER	MODE 23	2.12.1	INTEGER	MODE 23
2.13	INTEGER	MODE 24	2.13.1	INTEGER	MODE 24
2.14	INTEGER	MODE 25	2.14.1	INTEGER	MODE 25
2.15	INTEGER	MODE 26	2.15.1	INTEGER	MODE 26
2.16	INTEGER	MODE 27	2.16.1	INTEGER	MODE 27
2.17	INTEGER	MODE 28	2.17.1	INTEGER	MODE 28
2.18	INTEGER	MODE 29	2.18.1	INTEGER	MODE 29
2.19	INTEGER	MODE 30	2.19.1	INTEGER	MODE 30
2.20	INTEGER	MODE 31	2.20.1	INTEGER	MODE 31
2.21	INTEGER	MODE 32	2.21.1	INTEGER	MODE 32
2.22	INTEGER	MODE 33	2.22.1	INTEGER	MODE 33
2.23	INTEGER	MODE 34	2.23.1	INTEGER	MODE 34
2.24	INTEGER	MODE 35	2.24.1	INTEGER	MODE 35
2.25	INTEGER	MODE 36	2.25.1	INTEGER	MODE 36
2.26	INTEGER	MODE 37	2.26.1	INTEGER	MODE 37
2.27	INTEGER	MODE 38	2.27.1	INTEGER	MODE 38
2.28	INTEGER	MODE 39	2.28.1	INTEGER	MODE 39
2.29	INTEGER	MODE 40	2.29.1	INTEGER	MODE 40
2.30	INTEGER	MODE 41	2.30.1	INTEGER	MODE 41
2.31	INTEGER	MODE 42	2.31.1	INTEGER	MODE 42
2.32	INTEGER	MODE 43	2.32.1	INTEGER	MODE 43
2.33	INTEGER	MODE 44	2.33.1	INTEGER	MODE 44
2.34	INTEGER	MODE 45	2.34.1	INTEGER	MODE 45
2.35	INTEGER	MODE 46	2.35.1	INTEGER	MODE 46
2.36	INTEGER	MODE 47	2.36.1	INTEGER	MODE 47
2.37	INTEGER	MODE 48	2.37.1	INTEGER	MODE 48
2.38	INTEGER	MODE 49	2.38.1	INTEGER	MODE 49
2.39	INTEGER	MODE 50	2.39.1	INTEGER	MODE 50
2.40	INTEGER	MODE 51	2.40.1	INTEGER	MODE 51
2.41	INTEGER	MODE 52	2.41.1	INTEGER	MODE 52
2.42	INTEGER	MODE 53	2.42.1	INTEGER	MODE 53
2.43	INTEGER	MODE 54	2.43.1	INTEGER	MODE 54
2.44	INTEGER	MODE 55	2.44.1	INTEGER	MODE 55
2.45	INTEGER	MODE 56	2.45.1	INTEGER	MODE 56
2.46	INTEGER	MODE 57	2.46.1	INTEGER	MODE 57
2.47	INTEGER	MODE 58	2.47.1	INTEGER	MODE 58
2.48	INTEGER	MODE 59	2.48.1	INTEGER	MODE 59
2.49	INTEGER	MODE 60	2.49.1	INTEGER	MODE 60
2.50	INTEGER	MODE 61	2.50.1	INTEGER	MODE 61
2.51	INTEGER	MODE 62	2.51.1	INTEGER	MODE 62
2.52	INTEGER	MODE 63	2.52.1	INTEGER	MODE 63
2.53	INTEGER	MODE 64	2.53.1	INTEGER	MODE 64
2.54	INTEGER	MODE 65	2.54.1	INTEGER	MODE 65
2.55	INTEGER	MODE 66	2.55.1	INTEGER	MODE 66
2.56	INTEGER	MODE 67	2.56.1	INTEGER	MODE 67
2.57	INTEGER	MODE 68	2.57.1	INTEGER	MODE 68
2.58	INTEGER	MODE 69	2.58.1	INTEGER	MODE 69
2.59	INTEGER	MODE 70	2.59.1	INTEGER	MODE 70
2.60	INTEGER	MODE 71	2.60.1	INTEGER	MODE 71
2.61	INTEGER	MODE 72	2.61.1	INTEGER	MODE 72
2.62	INTEGER	MODE 73	2.62.1	INTEGER	MODE 73
2.63	INTEGER	MODE 74	2.63.1	INTEGER	MODE 74
2.64	INTEGER	MODE 75	2.64.1	INTEGER	MODE 75
2.65	INTEGER	MODE 76	2.65.1	INTEGER	MODE 76
2.66	INTEGER	MODE 77	2.66.1	INTEGER	MODE 77
2.67	INTEGER	MODE 78	2.67.1	INTEGER	MODE 78
2.68	INTEGER	MODE 79	2.68.1	INTEGER	MODE 79
2.69	INTEGER	MODE 80	2.69.1	INTEGER	MODE 80
2.70	INTEGER	MODE 81	2.70.1	INTEGER	MODE 81
2.71	INTEGER	MODE 82	2.71.1	INTEGER	MODE 82
2.72	INTEGER	MODE 83	2.72.1	INTEGER	MODE 83
2.73	INTEGER	MODE 84	2.73.1	INTEGER	MODE 84
2.74	INTEGER	MODE 85	2.74.1	INTEGER	MODE 85
2.75	INTEGER	MODE 86	2.75.1	INTEGER	MODE 86
2.76	INTEGER	MODE 87	2.76.1	INTEGER	MODE 87
2.77	INTEGER	MODE 88	2.77.1	INTEGER	MODE 88
2.78	INTEGER	MODE 89	2.78.1	INTEGER	MODE 89
2.79	INTEGER	MODE 90	2.79.1	INTEGER	MODE 90
2.80	INTEGER	MODE 91	2.80.1	INTEGER	MODE 91
2.81	INTEGER	MODE 92	2.81.1	INTEGER	MODE 92
2.82	INTEGER	MODE 93	2.82.1	INTEGER	MODE 93
2.83	INTEGER	MODE 94	2.83.1	INTEGER	MODE 94
2.84	INTEGER	MODE 95	2.84.1	INTEGER	MODE 95
2.85	INTEGER	MODE 96	2.85.1	INTEGER	MODE 96
2.86	INTEGER	MODE 97	2.86.1	INTEGER	MODE 97
2.87	INTEGER	MODE 98	2.87.1	INTEGER	MODE 98
2.88	INTEGER	MODE 99	2.88.1	INTEGER	MODE 99
2.89	INTEGER	MODE 100	2.89.1	INTEGER	MODE 100
2.90	INTEGER	MODE 101	2.90.1	INTEGER	MODE 101
2.91	INTEGER	MODE 102	2.91.1	INTEGER	MODE 102
2.92	INTEGER	MODE 103	2.92.1	INTEGER	MODE 103
2.93	INTEGER	MODE 104	2.93.1	INTEGER	MODE 104
2.94	INTEGER	MODE 105	2.94.1	INTEGER	MODE 105
2.95	INTEGER	MODE 106	2.95.1	INTEGER	MODE 106
2.96	INTEGER	MODE 107	2.96.1	INTEGER	MODE 107
2.97	INTEGER	MODE 108	2.97.1	INTEGER	MODE 108
2.98	INTEGER	MODE 109	2.98.1	INTEGER	MODE 109
2.99	INTEGER	MODE 110	2.99.1	INTEGER	MODE 110
3.00	INTEGER	MODE 111	3.00.1	INTEGER	MODE 111
3.01	INTEGER	MODE 112	3.01.1	INTEGER	MODE 112
3.02	INTEGER	MODE 113	3.02.1	INTEGER	MODE 113
3.03	INTEGER	MODE 114	3.03.1	INTEGER	MODE 114
3.04	INTEGER	MODE 115	3.04.1	INTEGER	MODE 115
3.05	INTEGER	MODE 116	3.05.1	INTEGER	MODE 116
3.06	INTEGER	MODE 117	3.06.1	INTEGER	MODE 117
3.07	INTEGER	MODE 118	3.07.1	INTEGER	MODE 118
3.08	INTEGER	MODE 119	3.08.1	INTEGER	MODE 119
3.09	INTEGER	MODE 120	3.09.1	INTEGER	MODE 120
3.10	INTEGER	MODE 121	3.10.1	INTEGER	MODE 121
3.11	INTEGER	MODE 122	3.11.1	INTEGER	MODE 122
3.12	INTEGER	MODE 123	3.12.1	INTEGER	MODE 123
3.13	INTEGER	MODE 124	3.13.1	INTEGER	MODE 124
3.14	INTEGER	MODE 125	3.14.1	INTEGER	MODE 125
3.15	INTEGER	MODE 126	3.15.1	INTEGER	MODE 126
3.16	INTEGER	MODE 127	3.16.1	INTEGER	MODE 127
3.17	INTEGER	MODE 128	3.17.1	INTEGER	MODE 128
3.18	INTEGER	MODE 129	3.18.1	INTEGER	MODE 129
3.19	INTEGER	MODE 130	3.19.1	INTEGER	MODE 130
3.20	INTEGER	MODE 131	3.20.1	INTEGER	MODE 131
3.21	INTEGER	MODE 132	3.21.1	INTEGER	MODE 132
3.22	INTEGER	MODE 133	3.22.1	INTEGER	MODE 133
3.23	INTEGER	MODE 134	3.23.1	INTEGER	MODE 134
3.24	INTEGER	MODE 135	3.24.1	INTEGER	MODE 135
3.25	INTEGER	MODE 136	3.25.1	INTEGER	MODE 136
3.26	INTEGER	MODE 137	3.26.1	INTEGER	MODE 137
3.27	INTEGER	MODE 138	3.27.1	INTEGER	MODE 138
3.28	INTEGER	MODE 139	3.28.1	INTEGER	MODE 139
3.29	INTEGER	MODE 140	3.29.1	INTEGER	MODE 140
3.30	INTEGER	MODE 141	3.30.1	INTEGER	MODE 141
3.31	INTEGER	MODE 142	3.31.1	INTEGER	MODE 142
3.32	INTEGER	MODE 143	3.32.1	INTEGER	MODE 143
3.33	INTEGER	MODE 144	3.33.1	INTEGER	MODE 144
3.34	INTEGER	MODE 145	3.34.1	INTEGER	MODE 145
3.35	INTEGER	MODE 146	3.35.1	INTEGER	MODE 146
3.36	INTEGER	MODE 147	3.36.1	INTEGER	MODE 147
3.37	INTEGER	MODE 148	3.37.1	INTEGER	MODE 148
3.38	INTEGER	MODE 149	3.38.1	INTEGER	MODE 149
3.39	INTEGER	MODE 150	3.39.1	INTEGER	MODE 150
3.40	INTEGER	MODE 151	3.40.1	INTEGER	MODE 151
3.41	INTEGER	MODE 152	3.41.1	INTEGER	MODE 152
3.42	INTEGER	MODE 153	3.42.1	INTEGER	MODE 153
3.43	INTEGER	MODE 154	3.43.1	INTEGER	MODE 154
3.44	INTEGER	MODE 155	3.44.1	INTEGER	MODE 155
3.45	INTEGER	MODE 156	3.45.1	INTEGER	MODE 156
3.46	INTEGER	MODE 157	3.46.1	INTEGER	MODE 157
3.47	INTEGER	MODE 158	3.47.1	INTEGER	MODE 158
3.48	INTEGER	MODE 159	3.48.1	INTEGER	MODE 159
3.49	INTEGER	MODE 160	3.49.1	INTEGER	MODE 160
3.50	INTEGER	MODE 161	3.50.1	INTEGER	MODE 161
3.51	INTEGER	MODE 162	3.51.1	INTEGER	MODE 162
3.52	INTEGER	MODE 163	3.52.1	INTEGER	MODE 163
3.53	INTEGER	MODE 164	3.53.1	INTEGER	MODE 164
3.54	INTEGER	MODE 165	3.54.1	INTEGER	MODE 165
3.55	INTEGER	MODE 166	3.55.1	INTEGER	MODE 166
3.56	INTEGER	MODE 167	3.56.1	INTEGER	MODE 167
3.57	INTEGER	MODE 168	3.57.1	INTEGER	MODE 168
3.58	INTEGER	MODE 169	3.58.1	INTEGER	MODE 169
3.59	INTEGER	MODE 170	3.59.1	INTEGER	MODE 170
3.60	INTEGER	MODE 171	3.60.1	INTEGER	MODE 171
3.61	INTEGER	MODE 172	3.61.1	INTEGER	MODE 172
3.62	INTEGER	MODE 173	3.62.1	INTEGER	MODE 173
3.63	INTEGER	MODE 174	3.63.1	INTEGER	MODE 174
3.64	INTEGER	MODE 175	3.64.1	INTEGER	MODE 175
3.65	INTEGER	MODE 176	3.65.1	INTEGER	MODE 176
3.66	INTEGER	MODE 177	3.66.1	INTEGER	MODE 177
3.67	INTEGER	MODE 178	3.67.1	INTEGER	MODE 178
3.68	INTEGER	MODE 179	3.68.1	INTEGER	MODE 179
3.69	INTEGER	MODE 180	3.69.1	INTEGER	MODE 180
3.70	INTEGER	MODE 181	3.70.1	INTEGER	MODE 181
3.71	INTEGER	MODE 182	3.71.1	INTEGER	MODE 182
3.72	INTEGER	MODE 183	3.72.1	INTEGER	MODE 183
3.73	INTEGER	MODE 184	3.73.1	INTEGER	MODE 184
3.74	INTEGER	MODE 185	3.74.1	INTEGER	MODE 185
3.75	INTEGER	MODE 186	3.75.1	INTEGER	MODE 186
3.76	INTEGER	MODE 187	3.76.1	INTEGER	MODE 187
3.77	INTEGER	MODE 188	3.77.1	INTEGER	MODE 188
3.78	INTEGER	MODE 189	3.78.1	INTEGER	MODE 189
3.79	INTEGER	MODE 190	3.79.1	INTEGER	MODE 190
3.80	INTEGER	MODE 191	3.80.1	INTEGER	MODE 191
3.81	INTEGER	MODE 192	3.81.1	INTEGER	MODE 192
3.82	INTEGER	MODE 193	3.82.1	INTEGER	MODE 193
3.83	INTEGER	MODE 194	3.83.1	INTEGER	MODE 194
3.84	INTEGER	MODE 195	3.84.1	INTEGER	MODE 19

LINE CACI SIMSCRIPT II.5 RELEASE NF 09/01/76 PAGE 22

```

1 EVENT T.IMPLEMENT.MOD
2 **
3 ** REPRESENTS DEVELOPMENT OF MODIFICATION BECAUSE OF FATIGUE TEST FAILURE
4 **
5 LET AMO = AMO + 1
6 LET FOCK = IOCK
7 LET REPAIR.MODIFIED = 0
8 LET TO.REPAIR.MODIFIED = IOCK
9 LET COST.OF.REPAIRS = 0.0
10 LET AIRFRAME.TIME = 0.0
11 LET ACTUAL.AVG.FAT.LIFE = PREDICTED.LIFE
12 IF MOD.TESTED = "NO"
13 LET AMO = AMO + ((1.0-MO.R)*.15)
14 LET NSIG = SIG.R + .45
15 CALL RECALC.LIFE(NMU,NSIG,PREDICTED.LIFE) YIELDING ACTUAL.AVG.FAT.LIFE
16 REGARDLESS
17 FOR EVERY AIRCRAFT IN ACTIVE.FLEET
18 DO
19 LET ID = TAIL.ID
20 IF IAAFL LT USAGE.LIFE
21 LET TMOO.PEVOTING(ID) = "YES"
22 IF ENTRY.TIME(AIRPLANE(ID)) + C7+IAAFL GT TIME.Y
23 SCHEDULE A T.TSPECTION.INCREASE(ID) AT ENTRY.TIME(AIRPLANE(ID)) + C7 +
24 IAAFL
25 LET ATII(ID) = T.INSPECTION.INCREASE
26 ALWAYS
27 REGARDLESS
28 LOOP
29 RETURN
30 END

```

LOCAL VARIABLES OF THIS ROUTINE

INTEGER	WORD	5	NMU	DOUBLE	WORD	1
DOUBLE	WORD	4				

L.9  
NSIG

09/01/76 PAGE 23

LINE CACT SIMSCRIPT IL5 RELEASE PF

```

1 ROUTINE PREDICT SERVICE DAMAGE YIELDING DMS.TU.SERVICE.DAMAGE
2 **
3 ** GENERATES TIME TO SERVICE DAMAGE OCCURRENCE FOR A GIVEN AIRCRAFT FROM A
4 ** CONSTANT SERVICE DAMAGE OCCURRENCE RATE
5 **
6 ** IF DMS.OCCURRANCE RATE LE 0.0
7   LET DMS.TU.SERVICE.DAMAGE = 2.0 * (DMS.LIFE
8   RETURN
9 ELSE
10  LET DMS.DAMAGE(2)
11  LET DMS.TU.SERVICE.DAMAGE = -LOG.F.F(2) / DMS.OCCURRENCE.RATE
12  RETURN
13

```

LOCAL VARIABLES OF THIS ROUTINE

DMS.TU.SERVICE.DAMAGE 1 20 000000 0000 5

LINE CACT SIMSCRIPT II.5 RELEASE OF 09/01/76 PAGE 24

```

1 ROUTINE PREICT.CORROSION YIELDING HOURS.TO.CORROSION
2 **
3 ** GENERATES TIME TO CORROSION INITIATION FROM ELEMENT HAZARD RATE WHICH IS
4 ** APPROXIMATED BY TWO CONSTANT OCCURRENCE RATES
5 **
6 ** 1.COR.OCCURRENCE.RATE = FIRST CONSTANT OCCURRENCE RATE
7 ** 2.COR.OCCURRENCE.RATE = SECOND CONSTANT OCCURRENCE RATE
8 ** COR.RATE.CHANGE = TIME OF AIRFRAME AT WHICH SECOND RATE IS USED
9 **
10 LET HN = MAXIMUM.I(4)
11 LET CACT = COR.RATE.CHANGE - TIME.V * ENTRY.TIME(AIRPLANE(10))
12 IF CACT LE 0.0
13 IF 2.COR.OCCURRENCE.RATE LE 0.0
14 LET MINUS.I(1).CORROSION = 2.0 * USAGE.LIFE
15 RETURN
16 ELSE
17 LET HOURS.TO.CORROSION = LOG.F.F(HN) / (-2.COR.OCCURRENCE.RATE)
18 RETURN
19 OTHERWISE
20 **
21 IF HN GE MAX.F(-CACT * 1.COR.OCCURRENCE.RATE)
22 IF 1.COR.OCCURRENCE.RATE LE 0.0
23 LET MINUS.I(1).CORROSION = 2.0 * USAGE.LIFE
24 RETURN
25 ELSE
26 LET MINUS.I(1).CORROSION = LOG.F.F(HN) / (-1.COR.OCCURRENCE.RATE)
27 RETURN
28 OTHERWISE
29 IF 2.COR.OCCURRENCE.RATE LE 0.0
30 LET MINUS.I(1).CORROSION = 2.0 * USAGE.LIFE
31 RETURN
32 ELSE
33 LET LN = (2.COR.OCCURRENCE.RATE - 1.COR.OCCURRENCE.RATE) * CACT
34 LET HOURS.TO.CORROSION = LOG.E.F(HN/EXP.F(LN)) / (-2.COR.OCCURRENCE.RATE)
35 RETURN
36 END

```

LOCAL VARIABLES OF THIS ROUTINE

NAME	TYPE	VALUE
CACT	DOUBLE	5
LN	DOUBLE	1
HN	DOUBLE	5

CACT  
LN

LINE CACT SIMSCRIPT ILS RELEASE AF 09/01/76 PAGE 25

```

1 EVENT COMBUSTION(LOC) SAVING THE EVENT NOTICE
2
3 ** REMOVED THE INITIATION OF CORROSION; REMAINING TIME TO CRACK INITIATION OF
4 ** ALL SCHEDULED CRACKS IS REDUCED BY COM.MULPLYING.FACTOR
5
6 DEFINE ILOC AS AN INTEGER VARIABLE
7 LET ILOC = ILOC
8 LET RST = (SACT.LIFE - TIME.V + ENTRY.TIME(AIRPLANE(LOC)))
9 LET VCMT = TIME.V - CTIME.TIME(LOC)
10 LET RICH = TIME.V - CTIME.TIME(LOC)
11 LET RICH = ILOC + 1
12 LET RICH = RICH + 1
13 LET CTIME(LOC) = "YES"
14 IF RICH(LOC) = "YES"
15 LET CTIME(LOC) = "YES"
16 IF RICH(LOC) = "YES"
17 LET CTIME(LOC) = "YES"
18 IF RICH(LOC) = "YES"
19 LET CTIME(LOC) = "YES"
20 IF RICH(LOC) = "YES"
21 LET CTIME(LOC) = "YES"
22 IF RICH(LOC) = "YES"
23 LET CTIME(LOC) = "YES"
24 IF RICH(LOC) = "YES"
25 LET CTIME(LOC) = "YES"
26 IF RICH(LOC) = "YES"
27 LET CTIME(LOC) = "YES"
28 IF RICH(LOC) = "YES"
29 LET CTIME(LOC) = "YES"
30 IF RICH(LOC) = "YES"
31 LET CTIME(LOC) = "YES"
32 IF RICH(LOC) = "YES"
33 LET CTIME(LOC) = "YES"
34 IF RICH(LOC) = "YES"
35 LET CTIME(LOC) = "YES"
36 IF RICH(LOC) = "YES"
37 LET CTIME(LOC) = "YES"
38 IF RICH(LOC) = "YES"
39 LET CTIME(LOC) = "YES"
40 IF RICH(LOC) = "YES"
41 LET CTIME(LOC) = "YES"
42 IF RICH(LOC) = "YES"
43 LET CTIME(LOC) = "YES"
44 IF RICH(LOC) = "YES"
45 LET CTIME(LOC) = "YES"
46 IF RICH(LOC) = "YES"
47 LET CTIME(LOC) = "YES"
48 IF RICH(LOC) = "YES"
49 LET CTIME(LOC) = "YES"
50 IF RICH(LOC) = "YES"

```



09/01/70 PAGE 26

```

LINE CACI S1500001 11.5 RELEASE RF
51 LET 2.ITE = ADE(ID)
52 LET TRI = (TIME.A(2.ITE)-TIME.V) / CGN1(ID)
53 CANCEL THE 2.ITE
54 IF TRI LT RST
55 RESCHEDULE THE 2.ITE AT TIME.V + TRI
56 JUMP AHEAD
57 ELSE
58 DESTROY THE 2.ITE
59 LET I2(ID) = "NO"
60 WHERE
61 IF I2(ID) = "YES"
62 LET 3.ITE = ARE(ID)
63 LET TRI = (TIME.A(3.ITE)-TIME.V) / CGN1(ID)
64 CANCEL THE 3.ITE
65 IF TRI LT RST
66 RESCHEDULE THE 3.ITE AT TIME.V + TRI
67 JUMP AHEAD
68 ELSE
69 DESTROY THE 3.ITE
70 LET I3(ID) = "NO"
71 WHERE
72 IF I3(ID) = "YES"
73 LET 4.ITE = ARE(ID)
74 LET TRI = (TIME.A(4.ITE)-TIME.V) / CGN1(ID)
75 CANCEL THE 4.ITE
76 IF TRI LT RST
77 RESCHEDULE THE 4.ITE AT TIME.V + TRI
78 JUMP AHEAD
79 ELSE
80 DESTROY THE 4.ITE
81 LET I4(ID) = "NO"
82 WHERE
83 IF I4(ID) = "YES"
84 LET 5.ITE = ARE(ID)
85 LET TRI = (TIME.A(5.ITE)-TIME.V) / CGN1(ID)
86 CANCEL THE 5.ITE
87 IF TRI LT RST
88 RESCHEDULE THE 5.ITE AT TIME.V + TRI
89 JUMP AHEAD
90 ELSE
91 DESTROY THE 5.ITE
92 LET I5(ID) = "NO"
93 WHERE
94 IF I5(ID) = "YES"
95 LET 6.ITE = ARE(ID)
96 LET TRI = (TIME.A(6.ITE)-TIME.V) / CGN1(ID)
97 CANCEL THE 6.ITE
98 IF TRI LT RST
99 RESCHEDULE THE 6.ITE AT TIME.V + TRI
100 JUMP AHEAD
101 ELSE
102 DESTROY THE 6.ITE
103 LET I6(ID) = "NO"
104 WHERE
105 IF I6(ID) = "YES"
106 LET 7.ITE = ARE(ID)
107 LET TRI = (TIME.A(7.ITE)-TIME.V) / CGN1(ID)
108 CANCEL THE 7.ITE
109 IF TRI LT RST
110 RESCHEDULE THE 7.ITE AT TIME.V + TRI
111 JUMP AHEAD
112 ELSE
113 DESTROY THE 7.ITE
114 LET I7(ID) = "NO"
115 WHERE
116 IF I7(ID) = "YES"
117 LET 8.ITE = ARE(ID)
118 LET TRI = (TIME.A(8.ITE)-TIME.V) / CGN1(ID)
119 CANCEL THE 8.ITE
120 IF TRI LT RST
121 RESCHEDULE THE 8.ITE AT TIME.V + TRI
122 JUMP AHEAD
123 ELSE
124 DESTROY THE 8.ITE
125 LET I8(ID) = "NO"
126 WHERE
127 IF I8(ID) = "YES"
128 LET 9.ITE = ARE(ID)
129 LET TRI = (TIME.A(9.ITE)-TIME.V) / CGN1(ID)
130 CANCEL THE 9.ITE
131 IF TRI LT RST
132 RESCHEDULE THE 9.ITE AT TIME.V + TRI
133 JUMP AHEAD
134 ELSE
135 DESTROY THE 9.ITE
136 LET I9(ID) = "NO"
137 WHERE
138 IF I9(ID) = "YES"
139 LET 10.ITE = ARE(ID)
140 LET TRI = (TIME.A(10.ITE)-TIME.V) / CGN1(ID)
141 CANCEL THE 10.ITE
142 IF TRI LT RST
143 RESCHEDULE THE 10.ITE AT TIME.V + TRI
144 JUMP AHEAD
145 ELSE
146 DESTROY THE 10.ITE
147 LET I10(ID) = "NO"
148 WHERE
149 IF I10(ID) = "YES"
150 LET 11.ITE = ARE(ID)
151 LET TRI = (TIME.A(11.ITE)-TIME.V) / CGN1(ID)
152 CANCEL THE 11.ITE
153 IF TRI LT RST
154 RESCHEDULE THE 11.ITE AT TIME.V + TRI
155 JUMP AHEAD
156 ELSE
157 DESTROY THE 11.ITE
158 LET I11(ID) = "NO"
159 WHERE
160 IF I11(ID) = "YES"
161 LET 12.ITE = ARE(ID)
162 LET TRI = (TIME.A(12.ITE)-TIME.V) / CGN1(ID)
163 CANCEL THE 12.ITE
164 IF TRI LT RST
165 RESCHEDULE THE 12.ITE AT TIME.V + TRI
166 JUMP AHEAD
167 ELSE
168 DESTROY THE 12.ITE
169 LET I12(ID) = "NO"
170 WHERE
171 IF I12(ID) = "YES"
172 LET 13.ITE = ARE(ID)
173 LET TRI = (TIME.A(13.ITE)-TIME.V) / CGN1(ID)
174 CANCEL THE 13.ITE
175 IF TRI LT RST
176 RESCHEDULE THE 13.ITE AT TIME.V + TRI
177 JUMP AHEAD
178 ELSE
179 DESTROY THE 13.ITE
180 LET I13(ID) = "NO"
181 WHERE
182 IF I13(ID) = "YES"
183 LET 14.ITE = ARE(ID)
184 LET TRI = (TIME.A(14.ITE)-TIME.V) / CGN1(ID)
185 CANCEL THE 14.ITE
186 IF TRI LT RST
187 RESCHEDULE THE 14.ITE AT TIME.V + TRI
188 JUMP AHEAD
189 ELSE
190 DESTROY THE 14.ITE
191 LET I14(ID) = "NO"
192 WHERE
193 IF I14(ID) = "YES"
194 LET 15.ITE = ARE(ID)
195 LET TRI = (TIME.A(15.ITE)-TIME.V) / CGN1(ID)
196 CANCEL THE 15.ITE
197 IF TRI LT RST
198 RESCHEDULE THE 15.ITE AT TIME.V + TRI
199 JUMP AHEAD
200 ELSE
201 DESTROY THE 15.ITE
202 LET I15(ID) = "NO"
203 WHERE
204 IF I15(ID) = "YES"
205 LET 16.ITE = ARE(ID)
206 LET TRI = (TIME.A(16.ITE)-TIME.V) / CGN1(ID)
207 CANCEL THE 16.ITE
208 IF TRI LT RST
209 RESCHEDULE THE 16.ITE AT TIME.V + TRI
210 JUMP AHEAD
211 ELSE
212 DESTROY THE 16.ITE
213 LET I16(ID) = "NO"
214 WHERE
215 IF I16(ID) = "YES"
216 LET 17.ITE = ARE(ID)
217 LET TRI = (TIME.A(17.ITE)-TIME.V) / CGN1(ID)
218 CANCEL THE 17.ITE
219 IF TRI LT RST
220 RESCHEDULE THE 17.ITE AT TIME.V + TRI
221 JUMP AHEAD
222 ELSE
223 DESTROY THE 17.ITE
224 LET I17(ID) = "NO"
225 WHERE
226 IF I17(ID) = "YES"
227 LET 18.ITE = ARE(ID)
228 LET TRI = (TIME.A(18.ITE)-TIME.V) / CGN1(ID)
229 CANCEL THE 18.ITE
230 IF TRI LT RST
231 RESCHEDULE THE 18.ITE AT TIME.V + TRI
232 JUMP AHEAD
233 ELSE
234 DESTROY THE 18.ITE
235 LET I18(ID) = "NO"
236 WHERE
237 IF I18(ID) = "YES"
238 LET 19.ITE = ARE(ID)
239 LET TRI = (TIME.A(19.ITE)-TIME.V) / CGN1(ID)
240 CANCEL THE 19.ITE
241 IF TRI LT RST
242 RESCHEDULE THE 19.ITE AT TIME.V + TRI
243 JUMP AHEAD
244 ELSE
245 DESTROY THE 19.ITE
246 LET I19(ID) = "NO"
247 WHERE
248 IF I19(ID) = "YES"
249 LET 20.ITE = ARE(ID)
250 LET TRI = (TIME.A(20.ITE)-TIME.V) / CGN1(ID)
251 CANCEL THE 20.ITE
252 IF TRI LT RST
253 RESCHEDULE THE 20.ITE AT TIME.V + TRI
254 JUMP AHEAD
255 ELSE
256 DESTROY THE 20.ITE
257 LET I20(ID) = "NO"
258 WHERE
259 IF I20(ID) = "YES"
260 LET 21.ITE = ARE(ID)
261 LET TRI = (TIME.A(21.ITE)-TIME.V) / CGN1(ID)
262 CANCEL THE 21.ITE
263 IF TRI LT RST
264 RESCHEDULE THE 21.ITE AT TIME.V + TRI
265 JUMP AHEAD
266 ELSE
267 DESTROY THE 21.ITE
268 LET I21(ID) = "NO"
269 WHERE
270 IF I21(ID) = "YES"
271 LET 22.ITE = ARE(ID)
272 LET TRI = (TIME.A(22.ITE)-TIME.V) / CGN1(ID)
273 CANCEL THE 22.ITE
274 IF TRI LT RST
275 RESCHEDULE THE 22.ITE AT TIME.V + TRI
276 JUMP AHEAD
277 ELSE
278 DESTROY THE 22.ITE
279 LET I22(ID) = "NO"
280 WHERE
281 IF I22(ID) = "YES"
282 LET 23.ITE = ARE(ID)
283 LET TRI = (TIME.A(23.ITE)-TIME.V) / CGN1(ID)
284 CANCEL THE 23.ITE
285 IF TRI LT RST
286 RESCHEDULE THE 23.ITE AT TIME.V + TRI
287 JUMP AHEAD
288 ELSE
289 DESTROY THE 23.ITE
290 LET I23(ID) = "NO"
291 WHERE
292 IF I23(ID) = "YES"
293 LET 24.ITE = ARE(ID)
294 LET TRI = (TIME.A(24.ITE)-TIME.V) / CGN1(ID)
295 CANCEL THE 24.ITE
296 IF TRI LT RST
297 RESCHEDULE THE 24.ITE AT TIME.V + TRI
298 JUMP AHEAD
299 ELSE
300 DESTROY THE 24.ITE
301 LET I24(ID) = "NO"
302 WHERE
303 IF I24(ID) = "YES"
304 LET 25.ITE = ARE(ID)
305 LET TRI = (TIME.A(25.ITE)-TIME.V) / CGN1(ID)
306 CANCEL THE 25.ITE
307 IF TRI LT RST
308 RESCHEDULE THE 25.ITE AT TIME.V + TRI
309 JUMP AHEAD
310 ELSE
311 DESTROY THE 25.ITE
312 LET I25(ID) = "NO"
313 WHERE
314 IF I25(ID) = "YES"
315 LET 26.ITE = ARE(ID)
316 LET TRI = (TIME.A(26.ITE)-TIME.V) / CGN1(ID)
317 CANCEL THE 26.ITE
318 IF TRI LT RST
319 RESCHEDULE THE 26.ITE AT TIME.V + TRI
320 JUMP AHEAD
321 ELSE
322 DESTROY THE 26.ITE
323 LET I26(ID) = "NO"
324 WHERE
325 IF I26(ID) = "YES"
326 LET 27.ITE = ARE(ID)
327 LET TRI = (TIME.A(27.ITE)-TIME.V) / CGN1(ID)
328 CANCEL THE 27.ITE
329 IF TRI LT RST
330 RESCHEDULE THE 27.ITE AT TIME.V + TRI
331 JUMP AHEAD
332 ELSE
333 DESTROY THE 27.ITE
334 LET I27(ID) = "NO"
335 WHERE
336 IF I27(ID) = "YES"
337 LET 28.ITE = ARE(ID)
338 LET TRI = (TIME.A(28.ITE)-TIME.V) / CGN1(ID)
339 CANCEL THE 28.ITE
340 IF TRI LT RST
341 RESCHEDULE THE 28.ITE AT TIME.V + TRI
342 JUMP AHEAD
343 ELSE
344 DESTROY THE 28.ITE
345 LET I28(ID) = "NO"
346 WHERE
347 IF I28(ID) = "YES"
348 LET 29.ITE = ARE(ID)
349 LET TRI = (TIME.A(29.ITE)-TIME.V) / CGN1(ID)
350 CANCEL THE 29.ITE
351 IF TRI LT RST
352 RESCHEDULE THE 29.ITE AT TIME.V + TRI
353 JUMP AHEAD
354 ELSE
355 DESTROY THE 29.ITE
356 LET I29(ID) = "NO"
357 WHERE
358 IF I29(ID) = "YES"
359 LET 30.ITE = ARE(ID)
360 LET TRI = (TIME.A(30.ITE)-TIME.V) / CGN1(ID)
361 CANCEL THE 30.ITE
362 IF TRI LT RST
363 RESCHEDULE THE 30.ITE AT TIME.V + TRI
364 JUMP AHEAD
365 ELSE
366 DESTROY THE 30.ITE
367 LET I30(ID) = "NO"
368 WHERE
369 IF I30(ID) = "YES"
370 LET 31.ITE = ARE(ID)
371 LET TRI = (TIME.A(31.ITE)-TIME.V) / CGN1(ID)
372 CANCEL THE 31.ITE
373 IF TRI LT RST
374 RESCHEDULE THE 31.ITE AT TIME.V + TRI
375 JUMP AHEAD
376 ELSE
377 DESTROY THE 31.ITE
378 LET I31(ID) = "NO"
379 WHERE
380 IF I31(ID) = "YES"
381 LET 32.ITE = ARE(ID)
382 LET TRI = (TIME.A(32.ITE)-TIME.V) / CGN1(ID)
383 CANCEL THE 32.ITE
384 IF TRI LT RST
385 RESCHEDULE THE 32.ITE AT TIME.V + TRI
386 JUMP AHEAD
387 ELSE
388 DESTROY THE 32.ITE
389 LET I32(ID) = "NO"
390 WHERE
391 IF I32(ID) = "YES"
392 LET 33.ITE = ARE(ID)
393 LET TRI = (TIME.A(33.ITE)-TIME.V) / CGN1(ID)
394 CANCEL THE 33.ITE
395 IF TRI LT RST
396 RESCHEDULE THE 33.ITE AT TIME.V + TRI
397 JUMP AHEAD
398 ELSE
399 DESTROY THE 33.ITE
400 LET I33(ID) = "NO"
401 WHERE
402 IF I33(ID) = "YES"
403 LET 34.ITE = ARE(ID)
404 LET TRI = (TIME.A(34.ITE)-TIME.V) / CGN1(ID)
405 CANCEL THE 34.ITE
406 IF TRI LT RST
407 RESCHEDULE THE 34.ITE AT TIME.V + TRI
408 JUMP AHEAD
409 ELSE
410 DESTROY THE 34.ITE
411 LET I34(ID) = "NO"
412 WHERE
413 IF I34(ID) = "YES"
414 LET 35.ITE = ARE(ID)
415 LET TRI = (TIME.A(35.ITE)-TIME.V) / CGN1(ID)
416 CANCEL THE 35.ITE
417 IF TRI LT RST
418 RESCHEDULE THE 35.ITE AT TIME.V + TRI
419 JUMP AHEAD
420 ELSE
421 DESTROY THE 35.ITE
422 LET I35(ID) = "NO"
423 WHERE
424 IF I35(ID) = "YES"
425 LET 36.ITE = ARE(ID)
426 LET TRI = (TIME.A(36.ITE)-TIME.V) / CGN1(ID)
427 CANCEL THE 36.ITE
428 IF TRI LT RST
429 RESCHEDULE THE 36.ITE AT TIME.V + TRI
430 JUMP AHEAD
431 ELSE
432 DESTROY THE 36.ITE
433 LET I36(ID) = "NO"
434 WHERE
435 IF I36(ID) = "YES"
436 LET 37.ITE = ARE(ID)
437 LET TRI = (TIME.A(37.ITE)-TIME.V) / CGN1(ID)
438 CANCEL THE 37.ITE
439 IF TRI LT RST
440 RESCHEDULE THE 37.ITE AT TIME.V + TRI
441 JUMP AHEAD
442 ELSE
443 DESTROY THE 37.ITE
444 LET I37(ID) = "NO"
445 WHERE
446 IF I37(ID) = "YES"
447 LET 38.ITE = ARE(ID)
448 LET TRI = (TIME.A(38.ITE)-TIME.V) / CGN1(ID)
449 CANCEL THE 38.ITE
450 IF TRI LT RST
451 RESCHEDULE THE 38.ITE AT TIME.V + TRI
452 JUMP AHEAD
453 ELSE
454 DESTROY THE 38.ITE
455 LET I38(ID) = "NO"
456 WHERE
457 IF I38(ID) = "YES"
458 LET 39.ITE = ARE(ID)
459 LET TRI = (TIME.A(39.ITE)-TIME.V) / CGN1(ID)
460 CANCEL THE 39.ITE
461 IF TRI LT RST
462 RESCHEDULE THE 39.ITE AT TIME.V + TRI
463 JUMP AHEAD
464 ELSE
465 DESTROY THE 39.ITE
466 LET I39(ID) = "NO"
467 WHERE
468 IF I39(ID) = "YES"
469 LET 40.ITE = ARE(ID)
470 LET TRI = (TIME.A(40.ITE)-TIME.V) / CGN1(ID)
471 CANCEL THE 40.ITE
472 IF TRI LT RST
473 RESCHEDULE THE 40.ITE AT TIME.V + TRI
474 JUMP AHEAD
475 ELSE
476 DESTROY THE 40.ITE
477 LET I40(ID) = "NO"
478 WHERE
479 IF I40(ID) = "YES"
480 LET 41.ITE = ARE(ID)
481 LET TRI = (TIME.A(41.ITE)-TIME.V) / CGN1(ID)
482 CANCEL THE 41.ITE
483 IF TRI LT RST
484 RESCHEDULE THE 41.ITE AT TIME.V + TRI
485 JUMP AHEAD
486 ELSE
487 DESTROY THE 41.ITE
488 LET I41(ID) = "NO"
489 WHERE
490 IF I41(ID) = "YES"
491 LET 42.ITE = ARE(ID)
492 LET TRI = (TIME.A(42.ITE)-TIME.V) / CGN1(ID)
493 CANCEL THE 42.ITE
494 IF TRI LT RST
495 RESCHEDULE THE 42.ITE AT TIME.V + TRI
496 JUMP AHEAD
497 ELSE
498 DESTROY THE 42.ITE
499 LET I42(ID) = "NO"
500 WHERE
501 IF I42(ID) = "YES"
502 LET 43.ITE = ARE(ID)
503 LET TRI = (TIME.A(43.ITE)-TIME.V) / CGN1(ID)
504 CANCEL THE 43.ITE
505 IF TRI LT RST
506 RESCHEDULE THE 43.ITE AT TIME.V + TRI
507 JUMP AHEAD
508 ELSE
509 DESTROY THE 43.ITE
510 LET I43(ID) = "NO"
511 WHERE
512 IF I43(ID) = "YES"
513 LET 44.ITE = ARE(ID)
514 LET TRI = (TIME.A(44.ITE)-TIME.V) / CGN1(ID)
515 CANCEL THE 44.ITE
516 IF TRI LT RST
517 RESCHEDULE THE 44.ITE AT TIME.V + TRI
518 JUMP AHEAD
519 ELSE
520 DESTROY THE 44.ITE
521 LET I44(ID) = "NO"
522 WHERE
523 IF I44(ID) = "YES"
524 LET 45.ITE = ARE(ID)
525 LET TRI = (TIME.A(45.ITE)-TIME.V) / CGN1(ID)
526 CANCEL THE 45.ITE
527 IF TRI LT RST
528 RESCHEDULE THE 45.ITE AT TIME.V + TRI
529 JUMP AHEAD
530 ELSE
531 DESTROY THE 45.ITE
532 LET I45(ID) = "NO"
533 WHERE
534 IF I45(ID) = "YES"
535 LET 46.ITE = ARE(ID)
536 LET TRI = (TIME.A(46.ITE)-TIME.V) / CGN1(ID)
537 CANCEL THE 46.ITE
538 IF TRI LT RST
539 RESCHEDULE THE 46.ITE AT TIME.V + TRI
540 JUMP AHEAD
541 ELSE
542 DESTROY THE 46.ITE
543 LET I46(ID) = "NO"
544 WHERE
545 IF I46(ID) = "YES"
546 LET 47.ITE = ARE(ID)
547 LET TRI = (TIME.A(47.ITE)-TIME.V) / CGN1(ID)
548 CANCEL THE 47.ITE
549 IF TRI LT RST
550 RESCHEDULE THE 47.ITE AT TIME.V + TRI
551 JUMP AHEAD
552 ELSE
553 DESTROY THE 47.ITE
554 LET I47(ID) = "NO"
555 WHERE
556 IF I47(ID) = "YES"
557 LET 48.ITE = ARE(ID)
558 LET TRI = (TIME.A(48.ITE)-TIME.V) / CGN1(ID)
559 CANCEL THE 48.ITE
560 IF TRI LT RST
561 RESCHEDULE THE 48.ITE AT TIME.V + TRI
562 JUMP AHEAD
563 ELSE
564 DESTROY THE 48.ITE
565 LET I48(ID) = "NO"
566 WHERE
567 IF I48(ID) = "YES"
568 LET 49.ITE = ARE(ID)
569 LET TRI = (TIME.A(49.ITE)-TIME.V) / CGN1(ID)
570 CANCEL THE 49.ITE
571 IF TRI LT RST
572 RESCHEDULE THE 49.ITE AT TIME.V + TRI
573 JUMP AHEAD
574 ELSE
575 DESTROY THE 49.ITE
576 LET I49(ID) = "NO"
577 WHERE
578 IF I49(ID) = "YES"
579 LET 50.ITE = ARE(ID)
580 LET TRI = (TIME.A(50.ITE)-TIME.V) / CGN1(ID)
581 CANCEL THE 50.ITE
582 IF TRI LT RST
583 RESCHEDULE THE 50.ITE AT TIME.V + TRI
584 JUMP AHEAD
585 ELSE
586 DESTROY THE 50.ITE
587 LET I50(ID) = "NO"
588 WHERE
589 IF I50(ID) = "YES"
590 LET 51.ITE = ARE(ID)
591 LET TRI = (TIME.A(51.ITE)-TIME.V) / CGN1(ID)
592 CANCEL THE 51.ITE
593 IF TRI LT RST
594 RESCHEDULE THE 51.ITE AT TIME.V + TRI
595 JUMP AHEAD
596 ELSE
597 DESTROY THE 51.ITE
598 LET I51(ID) = "NO"
599 WHERE
600 IF I51(ID) = "YES"
601 LET 52.ITE = ARE(ID)
602 LET TRI = (TIME.A(52.ITE)-TIME.V) / CGN1(ID)
603 CANCEL THE 52.ITE
604 IF TRI LT RST
605 RESCHEDULE THE 52.ITE AT TIME.V + TRI
606 JUMP AHEAD
607 ELSE
608 DESTROY THE 52.ITE
609 LET I52(ID) = "NO"
610 WHERE
611 IF I52(ID) = "YES"
612 LET 53.ITE = ARE(ID)
613 LET TRI = (TIME.A(53.ITE)-TIME.V) / CGN1(ID)
614 CANCEL THE 53.ITE
615 IF TRI LT RST
616 RESCHEDULE THE 53.ITE AT TIME.V + TRI
617 JUMP AHEAD
618 ELSE
619 DESTROY THE 53.ITE
620 LET I53(ID) = "NO"
621 WHERE
622 IF I53(ID) = "YES"
623 LET 54.ITE = ARE(ID)
624 LET TRI = (TIME.A(54.ITE)-TIME.V) / CGN1(ID)
625 CANCEL THE 54.ITE
626 IF TRI LT RST
627 RESCHEDULE THE 54.ITE AT TIME.V + TRI
628 JUMP AHEAD
629 ELSE
630 DESTROY THE 54.ITE
631 LET I54(ID) = "NO"
632 WHERE
633 IF I54(ID) = "YES"
634 LET 55.ITE = ARE(ID)
635 LET TRI = (TIME.A(55.ITE)-TIME.V) / CGN1(ID)
636 CANCEL THE 55.ITE
637 IF TRI LT RST
638 RESCHEDULE THE 55.ITE AT TIME.V + TRI
639 JUMP AHEAD
640 ELSE
641 DESTROY THE 55.ITE
642 LET I55(ID) = "NO"
643 WHERE
644 IF I55(ID) = "YES"
645 LET 56.ITE = ARE(ID)
646 LET TRI = (TIME.A(56.ITE)-TIME.V) / CGN1(ID)
647 CANCEL THE 56.ITE
648 IF TRI LT RST
649 RESCHEDULE THE 56.ITE AT TIME.V + TRI
650 JUMP AHEAD
651 ELSE
652 DESTROY THE 56.ITE
653 LET I56(ID) = "NO"
654 WHERE
655 IF I56(ID) = "YES"
656 LET 57.ITE = ARE(ID)
657 LET TRI = (TIME.A(57.ITE)-TIME.V) / CGN1(ID)
658 CANCEL THE 57.ITE
659 IF TRI LT RST
660 RESCHEDULE THE 57.ITE AT TIME.V + TRI
661 JUMP AHEAD
662 ELSE
663 DESTROY THE 57.ITE
664 LET I57(ID) = "NO"
665 WHERE
666 IF I57(ID) = "YES"
667 LET 58.ITE = ARE(ID)
668 LET TRI = (TIME.A(58.ITE)-TIME.V) / CGN1(ID)
669 CANCEL THE 58.ITE
670 IF TRI LT RST
671 RESCHEDULE THE 58.ITE AT TIME.V + TRI
672 JUMP AHEAD
673 ELSE
674 DESTROY THE 58.ITE
675 LET I58(ID) = "NO"
676 WHERE
677 IF I58(ID) = "YES"
678 LET 59.ITE = ARE(ID)
679 LET TRI = (TIME.A(59.ITE)-TIME.V) / CGN1(ID)
680 CANCEL THE 59.ITE
681 IF TRI LT RST
682 RESCHEDULE THE 59.ITE AT TIME.V + TRI
683 JUMP AHEAD
684 ELSE
685 DESTROY THE 59.ITE
686 LET I59(ID) = "NO"
687 WHERE
688 IF I59(ID) = "YES"
689 LET 60.ITE = ARE(ID)
690 LET TRI = (TIME.A(60.ITE)-TIME.V) / CGN1(ID)
691 CANCEL THE 60.ITE
692 IF TRI LT RST
693 RESCHEDULE THE 60.ITE AT TIME.V + TRI
694 JUMP AHEAD
695 ELSE
696 DESTROY THE 60.ITE
697 LET I60(ID) = "NO"
698 WHERE
699 IF I60(ID) = "YES"
700 LET 61.ITE = ARE(ID)
701 LET TRI = (TIME.A(61.ITE)-TIME.V) / CGN1(ID)
702 CANCEL THE 61.ITE
703 IF TRI LT RST
704 RESCHEDULE THE 61.ITE AT TIME.V + TRI
705 JUMP AHEAD
706 ELSE
707 DESTROY THE 61.ITE
708 LET I61(ID) = "NO"
709 WHERE
710 IF I61(ID) = "YES"
711 LET 62.ITE = ARE(ID)
712 LET TRI = (TIME.A(62.ITE)-TIME.V) / CGN1(ID)
713 CANCEL THE 62.ITE
714 IF TRI LT RST
715 RESCHEDULE THE 62.ITE AT TIME.V + TRI
716 JUMP AHEAD
717 ELSE
718 DESTROY THE 62.ITE
719 LET I62(ID) = "NO"
720 WHERE
721 IF I62(ID) = "YES"
722 LET 63.ITE = ARE(ID)
723 LET TRI = (TIME.A(63.ITE)-TIME.V) / CGN1(ID)
724 CANCEL THE 63.ITE
725 IF TRI LT RST
726 RESCHEDULE THE 63.ITE AT TIME.V + TRI
727 JUMP AHEAD
728 ELSE
729 DESTROY THE 63.ITE
730 LET I63(ID) = "NO"
731 WHERE
732 IF I63(ID) = "YES"
733 LET 64.ITE = ARE(ID)
734 LET TRI = (TIME.A(64.ITE)-TIME.V) / CGN1(ID)
735 CANCEL THE 64.ITE
736 IF TRI LT RST
737 RESCHEDULE THE 64.ITE AT TIME.V + TRI
738 JUMP AHEAD
739 ELSE
740 DESTROY THE 64.ITE
741 LET I64(ID) = "NO"
742 WHERE
743 IF I64(ID) = "YES"
744 LET 65.ITE = ARE(ID)
745 LET TRI = (TIME.A(65.ITE)-TIME.V) / CGN1(ID)
746 CANCEL THE 65.ITE
747 IF TRI LT RST
748 RESCHEDULE THE 65.ITE AT TIME.V + TRI
749 JUMP AHEAD
750 ELSE
751 DESTROY THE 65.ITE
752 LET I65(ID) = "NO"
753 WHERE
754 IF I65(ID) = "YES"
755 LET 66.ITE = ARE(ID)
756 LET TRI = (TIME.A(66.ITE)-TIME.V) / CGN1(ID)
757 CANCEL THE 66.ITE
758 IF TRI LT RST
759 RESCHEDULE THE 66.ITE AT TIME.V + TRI
760 JUMP AHEAD
761 ELSE
762 DESTROY THE 66.ITE
763 LET I66(ID) = "NO"
764 WHERE
765 IF I66(ID) = "YES"
766 LET 67.ITE = ARE(ID)
767 LET TRI = (TIME.A(67.ITE)-TIME.V) / CGN1(ID)
768 CANCEL THE 67.ITE
769 IF TRI LT RST
770 RESCHEDULE THE 67.ITE AT TIME.V + TRI
771 JUMP AHEAD
772 ELSE
773 DESTROY THE 67.ITE
774 LET I67(ID) = "NO"
775 WHERE
776 IF I67(ID) = "YES"
777 LET 68.ITE = ARE(ID)
778 LET TRI = (TIME.A(68.ITE)-TIME.V) / CGN1(ID)
779 CANCEL THE 68.ITE
780 IF TRI LT RST
781 RESCHEDULE THE 68.ITE AT TIME.V + TRI
782 JUMP AHEAD
783 ELSE
784 DESTROY THE 68.ITE
785 LET I68(ID) = "NO"
786 WHERE
787 IF I68(ID) = "YES"
788 LET 69.ITE = ARE(ID)
789 LET TRI = (TIME.A(69.ITE)-TIME.V) / CGN1(ID)
790 CANCEL THE 69.ITE
791 IF TRI LT RST
792 RESCHEDULE THE 69.ITE AT TIME.V + TRI
793 JUMP AHEAD
794 ELSE
795 DESTROY THE 69.ITE
796 LET I69(ID) = "NO"
797 WHERE
798 IF I69(ID) = "YES"
799 LET 70.ITE = ARE(ID)
800 LET TRI = (TIME.A(70.ITE)-TIME.V) / CGN1(ID)
801 CANCEL THE 70.ITE
802 IF TRI LT RST
803 RESCHEDULE THE 70.ITE AT TIME.V + TRI
804 JUMP AHEAD
805 ELSE
806 DESTROY THE 70.ITE
807 LET I70(ID) = "NO"
808 WHERE
809 IF I70(ID) = "YES"
810 LET 71.ITE = ARE(ID)
811 LET TRI = (TIME.A(71.ITE)-TIME.V) / CGN1(ID)
812 CANCEL THE 71.ITE
813 IF TRI LT RST
814 RESCHEDULE THE 71.ITE AT TIME.V + TRI
815 JUMP AHEAD
816 ELSE
817 DESTROY THE 71.ITE
818 LET I71(ID) = "NO"
819 WHERE
820 IF I71(ID) = "YES"
821 LET 72.ITE = ARE(ID)
822 LET TRI = (TIME.A(72.ITE)-TIME.V) / CGN1(ID)
823 CANCEL THE 72.ITE
824 IF TRI LT RST
825 RESCHEDULE THE 72.ITE AT TIME.V + TRI
826 JUMP AHEAD
827 ELSE
828 DESTROY THE 72.ITE
829 LET I72(ID) = "NO"
830 WHERE
831 IF I72(ID) = "YES"
832 LET 73.ITE = ARE(ID)
833 LET TRI = (TIME.A(73.ITE)-TIME.V) / CGN1(ID)
834 CANCEL THE 73.ITE
835 IF TRI LT RST
836 RESCHEDULE THE 73.ITE AT TIME.V + TRI
837 JUMP AHEAD
838 ELSE
839 DESTROY THE 73.ITE
840 LET I73(ID) = "NO"
841 WHERE
842 IF I73(ID) = "YES"
843 LET 74.ITE = ARE(ID)
844 LET TRI = (TIME.A(74.ITE)-TIME.V) / CGN1(ID)
845 CANCEL THE 74.ITE
846 IF TRI LT RST
847 RESCHEDULE THE 74.ITE AT TIME.V + TRI
848 JUMP AHEAD
849 ELSE
850 DESTROY THE 74.ITE
851 LET I74(ID) = "NO"
852 WHERE
853 IF I74(ID) = "YES"
854 LET 75.ITE = ARE(ID)
855 LET TRI = (TIME.A(75.ITE)-TIME.V) / CGN1(ID)
856 CANCEL THE 75.ITE
857 IF TRI LT RST
858 RESCHEDULE THE 75.ITE AT TIME.V + TRI
859 JUMP AHEAD
860 ELSE
861 DESTROY THE 75.ITE
862 LET I75(ID) = "NO"
863 WHERE
864 IF I75(ID) = "YES"
865 LET 76.ITE = ARE(ID)
866 LET TRI = (TIME.A(76.ITE)-TIME.V) / CGN1(ID)
867 CANCEL THE 76.ITE
868 IF TRI LT RST
869 RESCHEDULE THE 76.ITE AT TIME.V + TRI
870 JUMP AHEAD
871 ELSE
872 DESTROY THE 76.ITE
873 LET I76(ID) = "NO"
874 WHERE
875 IF I76(ID) = "YES"
876 LET 77.ITE = ARE(ID)
877 LET TRI = (TIME.A(77.ITE)-TIME.V) / CGN1(ID)
878 CANCEL THE 77.ITE
879 IF TRI LT RST
880 RESCHEDULE THE 77.ITE AT TIME.V + TRI
881 JUMP AHEAD
882 ELSE
883 DESTROY THE 77.ITE
884 LET I77(ID) = "NO"
885 WHERE
886 IF I77(ID) = "YES"
887 LET 78.ITE = ARE(ID)
888 LET TRI = (TIME.A(78.ITE)-TIME.V) / CGN1(ID)
889 CANCEL THE 78.ITE
890 IF TRI LT RST
891 RESCHEDULE THE 78.ITE AT TIME.V + TRI
892 JUMP AHEAD
893 ELSE
894 DESTROY THE 78.ITE
895 LET I78(ID) = "NO"
896 WHERE
897 IF I78(ID) = "YES"
898 LET 79.ITE = ARE(ID)
899 LET TRI = (TIME.A(79.ITE)-TIME.V) / CGN1(ID)
900 CANCEL THE 79.ITE
901 IF TRI LT RST
902 RESCHEDULE THE 79.ITE AT TIME.V + TRI
903 JUMP AHEAD
904 ELSE
905 DESTROY THE 79.ITE
906 LET I79(ID) = "NO"
907 WHERE
908 IF I79(ID) = "YES"
909 LET 80.ITE = ARE(ID)
910 LET TRI = (TIME.A(80.ITE)-TIME.V) / CGN1(ID)
911 CANCEL THE 80.ITE
912 IF TRI LT RST
913 RESCHEDULE THE 80.ITE AT TIME.V + TRI
914 JUMP AHEAD
915 ELSE
916 DESTROY THE 80.ITE
917 LET I80(ID) = "NO"
918 WHERE
919 IF I80(ID) = "YES"
920 LET 81.ITE = ARE(ID)
921 LET TRI = (TIME.A(81.ITE)-TIME.V) / CGN1(ID)
922 CANCEL THE 81.ITE
923 IF TRI LT RST
924 RESCHEDULE THE 81.ITE AT TIME.V + TRI
925 JUMP AHEAD
926 ELSE
927 DESTROY THE 81.ITE
928 LET I81(ID) = "NO"
929 WHERE
930 IF I81(ID) = "YES"
931 LET 82.ITE = ARE(ID)
932 LET TRI = (TIME.A(82.ITE)-TIME.V) / CGN1(ID)
933 CANCEL THE 82.ITE
934 IF TRI LT RST
935 RESCHEDULE THE 82.ITE AT TIME.V + TRI
936 JUMP AHEAD
937 ELSE
938 DESTROY THE 82.ITE
939 LET I82(ID) = "NO"
940 WHERE
941 IF I82(ID) = "YES"
942 LET 83.ITE = ARE(ID)
943 LET TRI = (TIME.A(83.ITE)-TIME.V) / CGN1(ID)
944 CANCEL THE 83.ITE
945 IF TRI LT RST
946 RESCHEDULE THE 83.ITE AT TIME.V + TRI
947 JUMP AHEAD
948 ELSE
949 DESTROY THE 83.ITE
950 LET I83(ID) = "NO"
951 WHERE
952 IF I83(ID) = "YES"
953 LET 84.ITE = ARE(ID)
954 LET TRI = (TIME.A(84.ITE)-TIME.V) / CGN1(ID)
955 CANCEL THE 84.ITE
956 IF TRI LT RST
957 RESCHEDULE THE 84.ITE AT TIME.V + TRI
958 JUMP AHEAD
959 ELSE
960 DESTROY THE 84.ITE
961 LET I84(ID) = "NO"
962 WHERE
963 IF I84(ID) = "YES"
964 LET 85.ITE = ARE(ID)
965 LET TRI = (TIME.A(85.ITE)-TIME.V) / CGN1(ID)
966 CANCEL THE 85.ITE
967 IF TRI LT RST
968 RESCHEDULE THE 85.ITE AT TIME.V + TRI
969 JUMP AHEAD
970 ELSE
971 DESTROY THE 85.ITE
972 LET I85(ID) = "NO"
973 WHERE
974 IF I85(ID) = "YES"
975 LET 86.ITE = ARE(ID)
976 LET TRI = (TIME.A(86.ITE)-TIME.V) / CGN1(ID)
977 CANCEL THE 86.ITE
978 IF TRI LT RST
979 RESCHEDULE THE 86.ITE AT TIME.V + TRI
980 JUMP AHEAD
981 ELSE
982 DESTROY THE 86.ITE
983 LET I86(ID) = "NO"
984 WHERE
985 IF I86(ID) = "YES"
986 LET 87.ITE = ARE(ID)
987 LET TRI = (TIME.A(87.ITE)-TIME.V) / CGN1(ID)
988 CANCEL THE 87.ITE
989 IF TRI LT RST
990 RESCHEDULE THE 87.ITE AT TIME.V + TRI
991 JUMP AHEAD
992 ELSE
993 DESTROY THE 87.ITE
994 LET I87(ID) = "NO"
995 WHERE
996 IF I87(ID) = "YES"
997 LET 88.ITE = ARE(ID)
998 LET TRI = (TIME.A(88.ITE)-TIME.V) / CGN1(ID)
999 CANCEL THE 88.ITE
1000 IF TRI LT RST
1001 RESCHEDULE THE 88.ITE AT TIME.V + TRI
1002 JUMP AHEAD
1003 ELSE
1004 DESTROY THE 88.ITE
1005 LET I88(ID) = "NO"
1006 WHERE
1007 IF I88(ID) = "YES"
1008 LET 89.ITE = ARE(ID)
1009 LET TRI = (TIME.A(89.ITE)-TIME.V) / CGN1(ID)
1010 CANCEL THE 89.ITE
1011 IF TRI LT RST
1012 RESCHEDULE THE 89.ITE AT TIME.V + TRI
1013 JUMP AHEAD
1014 ELSE
1015 DESTROY THE 89.ITE
1016 LET I89(ID) = "NO"
1017 WHERE
1018 IF I89(ID) = "YES"
1019 LET 90.ITE = ARE(ID)
1020 LET TRI = (TIME.A(90.ITE)-TIME.V) / CGN1(ID)
1021 CANCEL THE 90.ITE
1022 IF TRI LT RST
1023 RESCHEDULE THE 90.ITE AT TIME.V + TRI
1024 JUMP AHEAD
1025 ELSE
1026 DESTROY THE 90.ITE
1027 LET I90(ID) = "NO"
1028 WHERE
1029 IF I90(ID) = "YES"
1030 LET 91.ITE = ARE(ID)
1031 LET TRI = (TIME.A(91.ITE)-TIME.V) / CGN1(ID)
1032 CANCEL THE 91.ITE
1033 IF TRI LT RST
1034 RESCHEDULE THE 91.ITE AT TIME.V + TRI
1035 JUMP AHEAD
1036 ELSE
1037 DESTROY THE 91.ITE
1038 LET I91(ID) = "NO"
1039 WHERE
1040 IF I91(ID) = "YES"
1041 LET 92.ITE = ARE(ID)
1042 LET TRI = (TIME.A(92.ITE)-TIME.V) / CGN1(ID)
1043 CANCEL THE 92.ITE
1044 IF TRI LT RST
1045 RESCHEDULE THE 92.ITE AT TIME.V + TRI
1046 JUMP AHEAD
1047 ELSE
1048 DESTROY THE 92.ITE
1049 LET I92(ID) = "NO"
1050 WHERE
1051 IF I92(ID
```

09/01/76 PAGE 27

LINE CACT SIMSCHIPI II.5 RELEASE AF

```

103 **
104 ** TEST FOR SCHEDULED FIRST CRACK INITIATION
105 **
106 IF 1.CC.EXISTS(10) = "NS"
107 LET 1.STRENGTH.REDUCTION = AISC(10)
108 LET REMAINING.LIFE = TIME.A(1.STRENGTH.REDUCTION) - TIME.V
109 LET REDUCED.REMAINING.LIFE = REMAINING.LIFE * (CDM.MULTIPLYING.FACTOR
110 IF LIST = 1.0
111 PRINT 1 LINE WITH TIME.V+REDUCED.REMAINING.LIFE+ENTRY.TIME(AIMPLANE(10))
112 AS FOLLOWS
113 1ST CRACK INITIATION PROJECTED AT ***** FLIGHT HOURS
114 ALWAYS
115 CANCEL THE 1.STRENGTH.REDUCTION
116 DESTROY THE 1.STRENGTH.REDUCTION
117 LET 1.CC.EXISTS(10) = "NS"
118 IF REDUCED.REMAINING.LIFE LT 1ST OR SD.SCH(10) = "YES"
119 SCHEDULE A 1.STRENGTH.REDUCTION(10) AT TIME.V + REDUCED.REMAINING.LIFE
120 LET AISC(10) = 1.STRENGTH.REDUCTION
121 LET 1.CC.EXISTS(10) = "NS"
122 ALWAYS
123 MEGAWLESS
124 **
125 ** TEST FOR SCHEDULED SECOND CRACK INITIATION
126 **
127 IF 2.CC.EXISTS(10) = "NS"
128 LET 2.STRENGTH.REDUCTION = A2SR(10)
129 LET REMAINING.LIFE = TIME.A(2.STRENGTH.REDUCTION) - TIME.V
130 LET REDUCED.REMAINING.LIFE = REMAINING.LIFE * (CDM.MULTIPLYING.FACTOR
131 IF LIST = 1.0
132 PRINT 1 LINE WITH TIME.V+REDUCED.REMAINING.LIFE+ENTRY.TIME(AIMPLANE(10))
133 AS FOLLOWS
134 2ND CRACK INITIATION PROJECTED AT ***** FLIGHT HOURS
135 ALWAYS
136 CANCEL THE 2.STRENGTH.REDUCTION
137 DESTROY THE 2.STRENGTH.REDUCTION
138 LET 2.CC.EXISTS(10) = "NS"
139 IF REDUCED.REMAINING.LIFE LT 1ST OR SD.SCH(10) = "YES"
140 SCHEDULE A 2.STRENGTH.REDUCTION(10) AT TIME.V + REDUCED.REMAINING.LIFE
141 LET A2SR(10) = 2.STRENGTH.REDUCTION
142 LET 2.CC.EXISTS(10) = "NS"
143 ALWAYS
144 MEGAWLESS
145 **
146 ** TEST FOR SCHEDULED THIRD CRACK INITIATION
147 **
148 IF 3.CC.EXISTS(10) = "NS"
149 LET 3.STRENGTH.REDUCTION = A3SR(10)
150 LET REMAINING.LIFE = TIME.A(3.STRENGTH.REDUCTION) - TIME.V
151 LET REDUCED.REMAINING.LIFE = REMAINING.LIFE * (CDM.MULTIPLYING.FACTOR
152 IF LIST = 1.0
153 PRINT 1 LINE WITH TIME.V+REDUCED.REMAINING.LIFE+ENTRY.TIME(AIMPLANE(10))
154 AS FOLLOWS
155 3RD CRACK INITIATION PROJECTED AT ***** FLIGHT HOURS

```

09/01/76 PAGE 28

```

LINE CACTI STNSCRIPT II.5 RELEASE HF
153 ALWAYS
154 CANCEL THE 1,STRENGTH,REDUCTION
155 DESTRUY THE 1,STRENGTH,REDUCTION
156 LET 3,C4,EXIST(10) = "NO"
157 IF REDUCED,REMAINING,LIFE = "NO"
158 SCHEMULE 1,1,STRENGTH,REDUCTION,LI 4ST IN ST,SC(10) = "YES"
159 LET 4,SS-(10) = 3,STRENGTH,REDUCTION,LI 4ST IN ST,SC(10) = "YES"
160 LET 1,C4,EXIST(10) = "NO"
161 ALWAYS
162 RETURN
163 END

```

# LOCAL VARIABLES OF THIS ROUTINE

CONV. MULTIP	DOUBLE	WORD	5	1.1	INTEGER	WORD	15
I.2	INTEGER	WORD	14	I.3	INTEGER	WORD	15
I.3	INTEGER	WORD	1	J.1	INTEGER	WORD	16
K.1	INTEGER	WORD	14	K.2	INTEGER	WORD	19
K.3	INTEGER	WORD	20	L.4	INTEGER	WORD	21
L.4	INTEGER	WORD	9	L.6	INTEGER	WORD	22
LIST	DOUBLE	WORD	7	N.1	INTEGER	WORD	17
MTM	DOUBLE	WORD	25	N.1	DOUBLE	WORD	11
REDUCED,WF	DOUBLE	WORD	29	REMAINING,WF	DOUBLE	WORD	27
4ST	DOUBLE	WORD	3	REMAINING,WF	DOUBLE	WORD	23

09/01/76 PAGE 29

LINE CACI SYNOPSIS II.5 RELEASE AF

```

1 ROUTINE RATE(A,S,N)
2
3 ** GENERATES ELEMENT CRACK PROPAGATION RATES REFLECTING VARIATION IN MATERIAL
4 ** PROPERTIES AND LOAD ENVIRONMENT VARIATION
5
6 ** DISTRIBUTION OF CRACK PROPAGATION RATES IS REPRESENTED AS NORMAL
7 ** MEAN PROPAGATION RATE
8 ** S = STANDARD DEVIATION
9 ** N = RANDOM NUMBER STREAM
10
11 ** METHOD BASED ON APPROXIMATIONS IN C. HASTINGS, APPROXIMATIONS FOR DIGITAL
12 ** COMPUTERS
13
14 ** IF PROCEDURE YIELDS NEGATIVE RATE, RATE IS SET EQUAL TO MEAN MINUS FOUR
15 ** STANDARD DEVIATIONS
16
17 ** DEFINE Z AS AN INTEGER VARIABLE
18 LET Z = 1.0 - RANDOM.F(N)
19 LET Z = Z * N
20 IF Z < 0.5
21 LET Z = 1.0 - Z
22 REGAMBLESS
23 LET A = SINT.F((LOG.F.F(1.0/Z)))
24 LET G1 = 2.515517 * .0028688 * .0103280882
25 LET G2 = 1.0 + 1.4327888 * .140269682 * .0013084443
26 LET Z = A - G1/G2
27 IF Z < 0.5
28 LET Z = -Z
29 REGAMBLESS
30 LET Z = Z * S + A
31 IF Z < 0
32 LET Z = -Z * S + A
33 REGAMBLESS
34 RETURN WITH Z
35

```

LOCAL VARIABLES OF THIS ROUTINE

DOUBLE	WORD 13	G2	DOUBLE	WORD 15
DOUBLE	WORD 1	N	INTEGER	WORD 5
DOUBLE	WORD 7	A	DOUBLE	WORD 9
DOUBLE	WORD 3	A	DOUBLE	WORD 11
DOUBLE	WORD 17			

09/01/76 PAGE 30

```

LINE CACI SIMSCRIPT IL-5 RELEASE AF
1 EVENT 1- STRENGTH-REDUCTION(IDISH) SAVING THE EVENT NOTICE
2 **
3 ** REPRESENTS FIRST CRACK INITIATION
4 **
5 OFFICE IDISH AS AN INTEGER VARIABLE
6 LET ID = IDISH
7 LET TAV = EVERY.TIME(AIRPLANE(ID)) * USAGE.LIFE
8 IF LTAV = "YES"
9   FOR I = 1 TO WJAC(LOX)
10    DO
11     IF ID = 1 (IDCLOX, I)
12     LET LIST = 1.0
13     SKIP 1 (IDCLOX, I) LINE
14     PRINT 1 LINE WITH ID, TIME-V-TIME(AIRPLANE(ID)) AS FOLLOWS
15     A/C NO. *** EXPERIENCES 1ST CRACK INITIATION AT ***** HOURS
16     LEAVE
17   ELSE
18     LOOP
19   ALWAYS
20   LET ICMT = TIME.V - CRACK.TIME(ID)
21   LET SICM = TIME.V - CRACK.TIME(ID)
22   LET QICM = TICM + 1
23   LET GICM = QICM + 1
24   LET I.IV(I) = "NO"
25   IF WJAC(I) = 1.0 LE 1.0
26   LET I.IV(I) = "YES"
27   IF LIST = 1.0
28     PRINT 1 LINE AS FOLLOWS
29     CRACK INITIATES INTERNALLY
30   ALWAYS
31   LET I = (CRIT.CRM.LG1*WJAC(I))/(WSP(I)*CWM(I))
32   IF PCCL GT 1.0
33     LET I = CRIT.CRM.LG1/(WSP(I)*CWM(I)) + (CRIT.CRM.LG1*(WJAC(I)-1.0))/
34     (WSP(I)*CWM(I))
35   ALWAYS
36   IF TIME.V + 1 LT TAV ON CUMULATIVES(I) = "NS"
37   SCHEDULE A 1. (IF (ID) AT TIME.V + 1
38   LET AIC(I) = 1. (IF
39   LET IF(I) = "YES"
40   ALWAYS
41   RECALCULS
42   LET I.CM.EATIS(I) = "YES"
43   IF I.CM.SC(I) = "NO"
44   CALL INSPECTION.SCHEDULE(I)
45   RECALCULS
46   ** PREDICT LIFE 1: EST. ONE FROM FIRST CRACK INITIATION
47   **
48   LET S0 = 2.75
49   LET SP = 1.0
50   LET KKH(ID) = MAXIMUM.F(I)
51   LET LG = LOG.E.F(WJAC(I))
52   LET GM = WSP(I) * CWM(I)

```

09/01/76 PAGE 51

LINE CACTI SUBSCRIPT II.5 RELEASE AF

```

52 LET GR2 = MFR(ID) + CCR1(ID)
53 LET I1 = CRIT.CRM.LGT / GR1
54 LET T2 = T1 + (FSAF.LGT - CRIT.CRM.LGT) / GR2
55 LET M1 = ((SU-SF)*GR1) / FSAF.LGT
56 LET R2 = ((SU-SF)*GR2) / FSAF.LGT
57 LET R3 = GR2 / (LCMT.IN.FAILURE - FSAF.LGT)
58 ..
59 IF TIME.V + T2 LT TAN IN CO.EXISTS(ID) = "S"
60 SCHEDULE A RECH.FAIL.SAF.LGT(ID) AT TIME.V + T2
61 LET AFRSL(ID) = GRACH.FAIL.SAF.LGT
62 LET AIL(ID) = "YES"
63 REGARDLESS
64 LET K1 = A + EXP.F(SU * R)
65 LET K2 = A * R1
66 LET TTF = -LOG.E.F((-2)/K1)*LG + 1.0)/R2
67 IF TTF LE T1
68 JUMP AHEAD
69 ELSE
70 LET S1 = SU - R1 * T1
71 LET K3 = M * R2
72 LET LK5 = LOG.F.F(A) + (M * S1 + R * R2 * T1)
73 LET K4 = (AEXP.F(R2S1))/K3
74 LET K5 = (K1/R2)*((EXP.F(-K4*1011))-1.0)
75 LET TTF = (LK5 - LOG.F.F(K4S.F(K4S.F(K4S.F(LGCH2-(K4K13)-0.0))/K10
76 IF TTF LE T2
77 JUMP AHEAD
78 ELSE
79 LET K10 = M * R3
80 LET ARC = -0.5F*H43AT2
81 IF ARC LT -175.0
82 LET ANG = -175.0
83 ALWAYS
84 LET K11 = K10/(AEXP.F(ARC))
85 LET K12 = AEXP.F(R2SP)/K10
86 LET K13 = EXP.F(M2P211 - M42212) - 1.0
87 LET TTF = -LOG.F.F(K11*(LGK12-(K4K13)-0.0))/K10
88 HERE
89 IF LIST = 1.0
90 PRINT 1 LINE WITH TIME.V+TTF-FINITY.TTF(ZIMPLANE(ID)) AS FOLLWS
    ELEMENT FAILURE PROJECTION AT 00000 ELIGT MOUWS
91 ALWAYS
92 IF TIME.V + TTF LT TAN IN CO.EXISTS(ID) = "S"
93 SCHEDULE A FAILURE(ID) AT TIME.V + TTF
94 LET AF(ID) = FAILURE
95 LET FSM(ID) = "YES"
96 ALWAYS
97 RFTIM.
98 END

```

THIS PAGE IS BEST QUALITY PRACTICABLE  
FROM COPY FURNISHED TO DDQ

09/01/76 PAGE 32

LINE CACI SYMSRIPT II.5 RELEASE 6F

LOCAL VARIABLES OF THIS ROUTINE

ARG	WORD 41	GM1	WORD 29
GR2	DOUBLE	I.1	WORD 11
I.2	INTEGER	I.3	WORD 13
IDISH	INTEGER	J.1	WORD 14
K.1	INTEGER	K.2	WORD 17
K.3	INTEGER	K.4	WORD 19
K1	DOUBLE	K10	WORD 59
K11	DOUBLE	K12	WORD 65
K13	DOUBLE	K2	WORD 45
K4	DOUBLE	K4	WORD 55
K9	DOUBLE	L.4	WORD 7
LG	DOUBLE	LGR5	WORD 53
LIST	DOUBLE	N.1	WORD 15
M.1	DOUBLE	R1	WORD 37
M2	DOUBLE	R5	WORD 41
SF	DOUBLE	SU	WORD 23
S1	DOUBLE	T	WORD 21
TAR	DOUBLE	TIF	WORD 47
T1	DOUBLE	T2	WORD 35

09/01/76 PAGE 53

LINE CACI SIMSCRIPT ILS RELEASE AF

1 EVENT P-STRENGTH-REDUCTION(102SM) SAVING THE EVENT NOTICE

2 \*\* REPRESENTS SECOND CRACK INITIATION

3 \*\*

4 \*\*

5 \*\*

6 \*\*

7 \*\*

8 \*\*

9 \*\*

10 \*\*

11 \*\*

12 \*\*

13 \*\*

14 \*\*

15 \*\*

16 \*\*

17 \*\*

18 \*\*

19 \*\*

20 \*\*

21 \*\*

22 \*\*

23 \*\*

24 \*\*

25 \*\*

26 \*\*

27 \*\*

28 \*\*

29 \*\*

30 \*\*

31 \*\*

32 \*\*

33 \*\*

34 \*\*

35 \*\*

36 \*\*

37 \*\*

38 \*\*

39 \*\*

40 \*\*

41 \*\*

42 \*\*

43 \*\*

44 \*\*

45 \*\*

46 \*\*

47 \*\*

48 \*\*

49 \*\*

50 \*\*

51 \*\*



09/01/76 PAGE 34

LINE CACT SIMSCRIPT II.5 RELEASE AF

```

52 LET CL = CCL + ((CL-CCL)/MSR(ID))*MSR(ID)
53 JUMP AHEAD
54 OTHERWISE
55 JUMP AHEAD
56 OTHERWISE
57 LET DL = (TAC-TAI) + MSR(ID)
58 IF CCL LT DL
59 LET CL = CCL + ((DL-CCL)/MSR(ID))*MSR(ID) + ((TIME,V-TAC)*MSR(ID)*CGRI(ID)
60 JUMP AHEAD
61 OTHERWISE
62 LET CL = DL + ((TIME,V-TAC)*MSR(ID)*CGRI(ID)
63 IF CCL LT CL
64 LET CL = CCL + ((CL-CCL)/MSR(ID))*MSR(ID)
65 REGARDLESS
66 JUMP AHEAD
67 OTHERWISE
68 LET CL = (TIME,V-TAI) + MSR(ID)
69 IF CL GT CCL
70 LET CL = CCL + ((CL-CCL)/MSR(ID))*MSR(ID)
71 REGARDLESS
72 HERE
73 LET SU = 2.75 - (1.75 * CL) / FSAF.LGT
74 IF CL GT FSAF.LGT
75 LET SU = 1.0 - (CL-FSAF.LGT)/(LGHT.TN-FAILURE-FSAF.LGT)
76 REGARDLESS
77 ** PREDICT TIME TO FAILURE FROM SECOND CHECK INITIATION
78 **
79 **
80 LET T1 = 0.0
81 IF CL LT CCL
82 LET T1 = ((CL-CCL) / (MSR(ID) * CGRI(ID)))
83 LET GR1 = 2.0 * MSR(ID) * CGRI(ID)
84 IF TAI LE TIME,V + TCL
85 LET GR2 = (MSR(ID) + MSR(ID)) * CGRI(ID)
86 JUMP AHEAD
87 OTHERWISE
88 LET X(1) = 0.0
89 LET Y(1) = 0.0
90 LET X(2) = TCL-T1
91 LET Y(2) = CCL + X(2) * MSR(ID) * CGRI(ID) - T1 * MSR(ID) * CGRI(ID)
92 LET X(3) = TAI - TIME,V - T1
93 LET Y(3) = Y(2) + (2.0 * MSR(ID) * CGRI(ID))
94 LET N = 3
95 JUMP AHEAD
96 REGARDLESS
97 IF TAI LE TIME,V + TCL
98 LET GR1 = (MSR(ID) + MSR(ID)) * CGRI(ID)
99 LET GR2 = GR1
100 JUMP AHEAD
101 OTHERWISE
102 LET X(1) = 0.0
103 LET Y(1) = 0.0
104 LET X(2) = TCL

```

09/01/76 PAGE 35

LINE CACI SIMS-INT 11.5 RELEASE NF

```

105 LET V(1) = CCL + TCLAMER(ID)*CGM1(ID)
106 LET V(2) = TAD + TIME.V
107 LET V(3) = V(2) + (V(3)-TCL) + 2.0 * WFR(ID) + CGM1(ID)
108 LET W = 5
109 WEND
110 IF W GE 5
111 LET T(1) = 100.0
112 DIM T(2) = 2.1
113 W = 0
114 LET W(1) = 1.0
115 DIM W(2) = 1.1
116 W = 0
117 LET W(1) = W + W(1) + W(2)
118 LET W(2) = W + W(1) + W(2)
119 LET W(3) = W + W(1) + W(2)
120 LET W(4) = W + W(1) + W(2)
121 LET W(5) = W + W(1) + W(2)
122 LET W(6) = W + W(1) + W(2)
123 LET W(7) = W + W(1) + W(2)
124 LET W(8) = W + W(1) + W(2)
125 LET W(9) = W + W(1) + W(2)
126 LET W(10) = W + W(1) + W(2)
127 IF CL LF CCL
128 LET W(1) = 0
129 LET W(2) = 0
130 LET W(3) = 0
131 LET W(4) = 0
132 LET W(5) = 0
133 LET W(6) = 0
134 LET W(7) = 0
135 LET W(8) = 0
136 LET W(9) = 0
137 LET W(10) = 0
138 LET W(11) = 0
139 LET W(12) = 0
140 LET W(13) = 0
141 LET W(14) = 0
142 LET W(15) = 0
143 LET W(16) = 0
144 LET W(17) = 0
145 LET W(18) = 0
146 LET W(19) = 0
147 LET W(20) = 0
148 LET W(21) = 0
149 LET W(22) = 0
150 LET W(23) = 0
151 LET W(24) = 0
152 LET W(25) = 0
153 LET W(26) = 0
154 LET W(27) = 0
155 LET W(28) = 0
156 LET W(29) = 0
157 LET W(30) = 0

```

09/01/76 PAGE 36

LINE CACT SIMSCRIPT II.5 RELEASE RF.

```

158 JUMP AHEAD
159 ELSE
160 LET S1 = SU - R1 * T1
161 LET KQ = R * R2
162 LET LGRS = LOG.E.F(A) + (H * S1 + R * R2 * T1)
163 LET KR = (A*EXP.F(M*SI))/KQ
164 LET KQ = (K1/K2)*(EXP.F(-M*W1*T1)-1.0)
165 LET TTF = (LGRS - LOG.E.F(M*SI))/KQ
166 IF TTF LE T2
167 JUMP AHEAD
168 ELSE
169 LET K10 = M*Q5
170 LET AMG = M*SF+M*Q1+T2
171 IF AMG LT -175.0
172 LET AMG = -175.0
173 ALWAYS
174 LET K11 = K10/(A*EXP.F(AMG))
175 LET K12 = A*EXP.F(M*SI)/K10
176 LET K13 = EXP.F(M*Q2+T1 - M*Q2+T2) - 1.0
177 LET TTF = -LOG.F.F(K11*(LG*K12-(K*Q13)-Q9))/K10
178 MEME
179 IF LIST = 1.0
180 PRINT 1 LINE WITH TIME,V,TTF,FATRY,TIME(AIRPLANE(ID)) AS FOLLOWS
      ELEMENT FAILURE PROJECTED AT ***** FLIGHT HOURS
      ALWAYS
181 IF TIME.V + TTF LT IAR OR CO.FXISTS(ID) = "NS"
182 IF FSM(ID) = "YES"
183 LET FAILURE = AF(ID)
184 CANCEL THE FAILURE
185 RESCHEDULE THE FAILURE(ID) AT TIME.V + TTF
186 JUMP AHEAD
187 ELSE
188 SCHEDULE A FAILURE(ID) AT TIME.V + TTF
189 LET AF(ID) = FAILURE
190 LET FSM(ID) = "YES"
191 MEME
192 ALWAYS
193 RELEASE X(*), Y(*), A(*)
194 RETURN
195 END

```

LOCAL VARIABLES OF THIS ROUTINE

ARG	DOUBLE	WORD	RQ	CCL	DOUBLE	WORD	27
CL	DOUBLE	WORD	35	PL	DOUBLE	WORD	37
CS1	DOUBLE	WORD	45	Q12	DOUBLE	WORD	45
I-1	INTEGER	WORD	15	I-2	INTEGER	WORD	16
I-3	INTEGER	WORD	17	ID2SR	INTEGER	WORD	1
J-1	INTEGER	WORD	14	K-1	INTEGER	WORD	26
K-2	INTEGER	WORD	21	K-3	INTEGER	WORD	22
K-4	INTEGER	WORD	23	K1	DOUBLE	WORD	71
K10	DOUBLE	WORD	87	K11	DOUBLE	WORD	91

09/01/76 PAGE 37

LINE	CACI	SINSCN107	II.5	RELEASE	AF
K12	DDUMLF	WORD 93	K12		
K2	DDUMLF	WORD 73	K4		
K4	DDUMLF	WORD 43	K9		
L.4	INTEGEM	WORD 11	L6		
L.4	DDUMLF	WORD 41	LIST		
L.4	DDUMLF	WORD 47	K.1		
K.1	DDUMLF	WORD 15	K1		
K2	DDUMLF	WORD 44	K1		
SF	DDUMLF	WORD 61	S2		
S1	DDUMLF	WORD 30	S1		
T	DDUMLF	WORD 24	TAC		
TAM	DDUMLF	WORD 7	TAT		
TCL	DDUMLF	WORD 51	TIF		
T1	DDUMLF	WORD 41	T2		
K15	WEAL	WORD 5	K1		
K1	DDUMLF	WORD 55	K1		
K1	DDUMLF	WORD 53	K1		
K1	WEAL	WORD 4	K1		

DDUMLF	WORD 95
DDUMLF	WORD 79
DDUMLF	WORD 85
DDUMLF	WORD 69
DDUMLF	WORD 9
INTEGEM	WORD 19
DDUMLF	WORD 63
DDUMLF	WORD 67
DDUMLF	WORD 57
DDUMLF	WORD 77
DDUMLF	WORD 55
DDUMLF	WORD 29
DDUMLF	WORD 75
DDUMLF	WORD 59
DDUMLF	WORD 49
DDUMLF	WORD 51
WEAL	WORD 3

09/01/76 PAGE 34

LINE CACT SIMSCRIPT II.5 RELEASE 8F

1 EVENT 3-STRENGTH-REDUCTION(IDSR) SAVING THE EVENT NOTICE

2 \*\* REPRESENTS THIRD CRACK INITIATION

3 \*\*

4 DEFINE IDSR AS AN INTEGER VARIABLE

5 DEFINE K, Y, X AS REAL, 1-DIMENSIONAL ARRAYS

6 RESERVE X(1), Y(1), X(2) AS 5

7 LET ID = IDSR

8 LET TAP = ENTRY.TIME(AIRPLANE(ID)) + USAGE.LIFE

9 IF LTIM = "YES"

10 FOR I = 1 TO MAX(LDA)

11 DO

12 IF ID = TLID(LDA,I)

13 LET LIST = 1.0

14 SKIP 1 OUTPUT LINE

15 PRINT 1 LINE WITH ID, TIME, V-ENTRY, TIME(AIRPLANE(ID)) AS FOLLOWS

16 A/C 3). \*\*\* EXPERIENCES 3RD CRACK INITIATION AT \*\*\*\*\* HOURS

17 LEAVE

18 ELSE

19 LOOP

20 ALWAYS

21 LET 3.INT(ID) = "NO"

22 IF RANDOM.F(10) LE 1.0 THEN

23 LET 3.INT(ID) = "YES"

24 IF LIST = 1.0

25 PRINT 1 LINE AS FOLLOWS

26 CRACK INITIATES INTERNALLY

27 ALWAYS

28 LET T = (CRIT.CRK.LGT\*PCLL)/(USR(ID)\*CGW(ID))

29 IF PCLL GT 1.0

30 LET T = CRIT.CRK.LGT/(MSW(ID)\*CGW(ID)) + (CRIT.CRK.LGT\*(PCLL-1.0))/

31 (MSW(ID)\*CGW(ID))

32 ALWAYS

33 IF TIME.V + T LT TAP OR CO-EXISTS(ID) = "NS"

34 SCHEDULE A 3.ITE(10) AT TIME.V + T

35 LET ASE(ID) = 3.ITE

36 LET TES(ID) = "YES"

37 ALWAYS

38 IF RANDOMLESS

39 LET 3.CO-EXISTS(ID) = "YES"

40 LET 1-STRENGTH-REDUCTION = AISR(ID)

41 LET 2-STRENGTH-REDUCTION = A2SR(ID)

42 LET CEL = CRIT.CRK.LGT

43 LET TCL = CEL/(MSW(ID)\*CGW(ID))

44 LET TAI = TIME.A(1-STRENGTH-REDUCTION)

45 LET TAP = TIME.A(2-STRENGTH-REDUCTION)

46 \*\* INTERVIEW PERSONAL STRENGTH REDUCTION \*\* (CAUSE OF FIRST AND SECOND CRACKS)

47 \*\*

48 IF CO-EXISTS(ID) = "YES"

49 LET CORROSION = AC(ID)

50 LET TAC = TIME.A(CORROSION)

51 IF TAC LT TAI

09/01/76 PAGE 39

```

LINE CACI SIMSCRIPT 11.5 RELEASE MF
52 LET ICL = (TIME.V-TA1) * MSW(ID) * CGM(ID)
53 LET PCL = (TIME.V-TA2) * MSW(ID) * CGM(ID)
54 IF ICL GT CCL
55 LET ICL = CCL + ((ICL-CCL)/MSW(ID))*MSW(ID)
56 *FLAGDLESS
57 IF PCL GT CCL
58 LET PCL = CCL + ((PCL-CCL)/MSW(ID))*MSW(ID)
59 *FLAGDLESS
60 LET CL = ICL + PCL
61 JUMP AMEAD
62 *TIMEBASE
63 IF TAC LT TA2
64 LET NL = (TAC-TA1) * ASW(ID)
65 LET ICL = CL + (TIME.V-TAC)*MSW(ID)*CGM(ID)
66 LET PCL = (TIME.V-TA2) * ASW(ID) * CGM(ID)
67 IF CCL LT NL
68 LET PCL = (TIME.V-TA2) * ASW(ID) * CGM(ID)
69 *FLAGDLESS
70 IF PCL GT CCL
71 LET PCL = CCL + ((PCL-CCL)/MSW(ID))*MSW(ID)
72 *FLAGDLESS
73 *FLAGDLESS
74 IF TAC LT TA1
75 LET ICL = (TIME.V-TA1) * MSW(ID) * CGM(ID)
76 IF ICL GT CCL
77 LET ICL = CCL + ((ICL-CCL)/MSW(ID))*MSW(ID)
78 *FLAGDLESS
79 *FLAGDLESS
80 *TIMEBASE
81 IF TAC LT TA1
82 LET ICL = (TIME.V-TA1) * MSW(ID)
83 IF ICL GT CCL
84 LET ICL = CCL + ((ICL-CCL)/MSW(ID))*MSW(ID)
85 *FLAGDLESS
86 LET PCL = (TIME.V-TA2) * MSW(ID)
87 IF PCL GT CCL
88 LET PCL = CCL + ((PCL-CCL)/MSW(ID))*MSW(ID)
89 *FLAGDLESS
90 *FLAGDLESS
91 LET CL = ICL + PCL
92 LET SW = 2.75 * (CL * CL) / FSAF.LGT
93 IF CL GT FSAF.LGT
94 LET SW = 1.75 * (CL-FSAF.LGT)/(LGT-FSAF.LGT)
95 *FLAGDLESS
96
97 ** PREDICT TIME TO FAILURE FROM INTER CHACK INITIATION.
98
99 LET T1 = 1.0
100 IF ICL LT CCL
101 LET T1 = (CCL-ICL) / (MSW(ID)*CGM(ID))
102 LET T2 = 1.0 * MSW(ID) * CGM(ID)
103 LET T2 = (CCL-PCL) / (MSW(ID)*CGM(ID))
104 IF T1 LT TIME.V + T2

```

09/31/75 PAGE 08

```

LINE  CACT ST-SC-1PT II.5  RELEASE AF
105  LET GR2 = (WER(10) + 2.0*MSW(10)) * CGW(10)
106  JUMP AMEAD
107  OTHERWISE
108  LET X(1) = 0.0
109  LET Y(1) = 0.0
110  LET X(2) = T2-T1
111  LET Y(2) = X(2) * (WER(10) + 2.0*MSW(10)) * CGW(10)
112  IF TAR LE TIME.V + TCL
113  LET X(3) = TAR - TIME.V - T1
114  LET Y(3) = Y(2) + (X(3)-X(2)) * (2.0*WER(10) + MSW(10)) * CGW(10)
115  LET N = 3
116  JUMP AMEAD
117  OTHERWISE
118  LET X(3) = TCL - T1
119  LET N = X(3) - T2 + T1
120  LET Y(3) = (X(3)*X(3))*WER(10)*CGW(10) + (T2-T1+X(3))*MSW(10)*CGW(10)
121  LET X(4) = TAR - TIME.V - T1
122  LET Y(4) = (X(4)-X(3)) * 3.0 * WER(10) * CGW(10)
123  LET N = 4
124  JUMP AMEAD
125  OTHERWISE
126  LET X(1) = 0.0
127  LET Y(1) = 0.0
128  IF >CL LT CCL
129  LET T2 = (CCL-2CL)/(MSR(10)*CGW(10))
130  IF TAR LE TIME.V + T2
131  LET GR2 = (2.0 * MSR(10) + WER(10)) * CGW(10)
132  LET GR1 = GR2
133  JUMP AMEAD
134  OTHERWISE
135  IF TAR LE TIME.V + TCL
136  LET X(2) = T2
137  LET Y(2) = T2 * (2.0 * MSW(10) + WER(10)) * CGW(10)
138  LET X(3) = TAR - TIME.V
139  LET Y(3) = Y(2) + (X(3)-X(2)) * (2.0 * WER(10) + MSW(10)) * CGW(10)
140  LET N = 3
141  JUMP AMEAD
142  OTHERWISE
143  LET X(2) = T2
144  LET Y(2) = T2 * (2.0 * MSW(10) + WER(10)) * CGW(10)
145  LET X(3) = TCL
146  LET Y(3) = Y(2) + (TCL-T2) * (2.0 * WER(10) + MSW(10)) * CGW(10)
147  LET X(4) = TAR - TIME.V
148  LET Y(4) = Y(3) + (X(4)-TCL) * (3.0 * WER(10) + CGW(10))
149  LET N = 4
150  JUMP AMEAD
151  OTHERWISE
152  IF TAR LE TIME.V + TCL
153  LET GR2 = (2.0 * WER(10) + MSW(10)) * CGW(10)
154  LET GR1 = GR2
155  JUMP AMEAD
156  OTHERWISE
157  LET X(2) = TCL

```

09/01/76 PAGE 41

LINE CACI SIMSCRIPT II.5 RELEASE RF

```

154 LET V(2) = TOL * (2.0 * WFR(ID) + MSR(ID)) * CGMI(ID)
155 LET X(3) = TAR - TIME.V
156 LET V(3) = V2 + (X(3)-X(2)) * (3.0 * WFR(ID)) * CGMI(ID)
157 LET V1 = V
158 NAME
159 IF N.GE 3
160 LET A(1) = 100.0
161 FOR I = 1 TO N
162 DO
163 LET A(I) = 1.0
164 LET I = 1 TO N
165 DO
166 LET A(I) = 1.0
167 LET I = 1 TO N
168 DO
169 LET A(I) = 1.0
170 LET I = 1 TO N
171 LET A(I) = 1.0
172 LET A(I) = 1.0
173 LET A(I) = 1.0
174 LET A(I) = 1.0
175 LET A(I) = 1.0
176 LET A(I) = 1.0
177 LET A(I) = 1.0
178 LET A(I) = 1.0
179 LET A(I) = 1.0
180 LET A(I) = 1.0
181 LET A(I) = 1.0
182 LET A(I) = 1.0
183 LET A(I) = 1.0
184 LET A(I) = 1.0
185 LET A(I) = 1.0
186 LET A(I) = 1.0
187 LET A(I) = 1.0
188 LET A(I) = 1.0
189 LET A(I) = 1.0
190 LET A(I) = 1.0
191 LET A(I) = 1.0
192 LET A(I) = 1.0
193 LET A(I) = 1.0
194 LET A(I) = 1.0
195 LET A(I) = 1.0
196 LET A(I) = 1.0
197 LET A(I) = 1.0
198 LET A(I) = 1.0
199 LET A(I) = 1.0
200 LET A(I) = 1.0
201 LET A(I) = 1.0
202 LET A(I) = 1.0
203 LET A(I) = 1.0
204 LET A(I) = 1.0
205 LET A(I) = 1.0
206 LET A(I) = 1.0
207 LET A(I) = 1.0
208 LET A(I) = 1.0
209 LET A(I) = 1.0
210 LET A(I) = 1.0

```



09/01/76 PAGE 42

LINE CACI SCRIPT II.5 PLEASE MF

```

211 JUMP AHEAD
212 ELSE
213 LET S1 = SU - R1 + T1
214 LET K4 = R + K2
215 LET LK4 = LOG.E.F(A) + (R * S1) + M * K2 + T1
216 LET K4 = (A*EXP.F(K4*S1))/K4
217 LET K9 = (K1/K2)*(EXP.F(M*K1*T1)-1.0)
218 LET ITF = (LK4 - LOG.F.F(AHS.F(K9)) - LOG.E.F(AHS.F(K9)))/K4
219 IF ITF LT T2
220 JUMP AHEAD
221 ELSE
222 LET K10 = K45
223 LET ARG = M*EXP.M*3*T2
224 IF ARG LT -175.0
225 LET ARG = -175.0
226 ALWAYS
227 LET K11 = K10/(A*EXP.F(ARG))
228 LET K12 = A*EXP.F(M*S1/K10)
229 LET K13 = EXP.F(M*K2*T1 - M*K2*T2) - 1.0
230 LET ITF = -LOG.E.F(K11*(LOG.F(K12*(M*K13)-K9)))/K10
231 MEME
232 IF LIST = 1.0
233 PRINT 1 LINE WITH TIME,V,TIME,F,TIME(ATWPLANE(ID)) AS FOLLOWS
      FLIGHT FAILURE PROJECTED AT ***** FLIGHT HOURS
234 ALWAYS
235 IF TIME,V + ITF LT T4 (M COEFFICIENTS) = "NS"
236 IF FSH(ID) = "YES"
237 LET FAILURE = AF(ID)
238 CANCEL THE FAILURE
239 RESCHEDULE THE FAILURE(ID) AT TIME,V + ITF
240 JUMP AHEAD
241 ELSE
242 RESCHEDULE A FAILURE(ID) AT TIME,V + ITF
243 LET AF(ID) = FAILURE
244 LET FSH(ID) = "YES"
245 MEME
246 ALWAYS
247 RELEASE X(*), Y(*), Z(*)
248 RETURN
249 END

```

LOCAL VARIABLES OF THIS MODULE

ARG	DOUBLE	WORD 99	CCL	DOUBLE	WORD 27
CL	DOUBLE	WORD 41	DL	DOUBLE	WORD 45
GM1	DOUBLE	WORD 39	GM2	DOUBLE	WORD 53
I.1	INTEGER	WORD 15	I.2	INTEGER	WORD 16
I.3	INTEGER	WORD 17	I.4	INTEGER	WORD 1
J.1	INTEGER	WORD 14	K.1	INTEGER	WORD 26
K.2	INTEGER	WORD 21	K.3	INTEGER	WORD 22
K.4	INTEGER	WORD 23	K1	DOUBLE	WORD 79
K10	DOUBLE	WORD 97	K11	DOUBLE	WORD 101

THIS PAGE IS BEST QUALITY PRACTICABLE  
FROM COPY FURNISHED TO DDG

09/01/74 PAGE 43

LINE	CACI	SIMSCRIPT	II.5	RELEASE	AF
K12	DOUBLE	WORD103		K13	
K2	DOUBLE	WORD A1		K4	
KA	DOUBLE	WORD 93		K9	
L.4	INTEGER	WORD 11		LG	
LKNS	DOUBLE	WORD 91		LIST	
N	DOUBLE	WORD 55		R.1	
R.1	DOUBLE	WORD 13		R1	
R2	DOUBLE	WORD 75		R5	
SF	DOUBLE	WORD 71		SPN	
SU	DOUBLE	WORD 45		SI	
T	DOUBLE	WORD 25		TAC	
TAR	DOUBLE	WORD 7		TAL	
TA2	DOUBLE	WORD 33		TCL	
TTF	DOUBLE	WORD A5		T1	
T2	DOUBLE	WORD 51		N	
AA	DOUBLE	WORD 61		AS	
AY	DOUBLE	WORD 63		AT	
A	REAL	WORD 3		KK	
V	REAL	WORD 4		V2	
ICL	DOUBLE	WORD 37		ZCL	

DOUBLE	WORD 105
DOUBLE	WORD 89
DOUBLE	WORD 95
DOUBLE	WORD A3
DOUBLE	WORD 9
DOUBLE	WORD 19
INTEGER	WORD 73
DOUBLE	WORD 77
DOUBLE	WORD 69
DOUBLE	WORD 87
DOUBLE	WORD 35
DOUBLE	WORD 31
DOUBLE	WORD 29
DOUBLE	WORD 87
DOUBLE	WORD 5
REAL	WORD 67
DOUBLE	WORD 85
DOUBLE	WORD 57
DOUBLE	WORD 59
DOUBLE	WORD 39

```

1  CACI DEMOSCRPT II-5  WELFAS HP
2  EVENT 1-11(EDITE)
3  DEFINE TIME AS AN INTEGER VARIABLE
4  LET I0 = TIME
5  LET I-1(EDITE) = "END"
6  LET IF(I0) = "END"
7  IF (TIME) = "END"
8  FOR I = 1 TO 1000(1000)
9  IF I = 1000(1000)
10  STOP 1-11(EDITE)
11  LET I-1(EDITE) = "END"
12  LET I-1(EDITE) = "END"
13  LET I-1(EDITE) = "END"
14  LET I-1(EDITE) = "END"
15  LET I-1(EDITE) = "END"
16  LET I-1(EDITE) = "END"
17  LET I-1(EDITE) = "END"
18  LET I-1(EDITE) = "END"
19  LET I-1(EDITE) = "END"
20  LET I-1(EDITE) = "END"
21  LET I-1(EDITE) = "END"
22  LET I-1(EDITE) = "END"
23  LET I-1(EDITE) = "END"
24  LET I-1(EDITE) = "END"
25  LET I-1(EDITE) = "END"
26  LET I-1(EDITE) = "END"
27  LET I-1(EDITE) = "END"
28  LET I-1(EDITE) = "END"
29  LET I-1(EDITE) = "END"
30  LET I-1(EDITE) = "END"
31  LET I-1(EDITE) = "END"
32  LET I-1(EDITE) = "END"
33  LET I-1(EDITE) = "END"
34  LET I-1(EDITE) = "END"
35  LET I-1(EDITE) = "END"
36  LET I-1(EDITE) = "END"
37  LET I-1(EDITE) = "END"
38  LET I-1(EDITE) = "END"
39  LET I-1(EDITE) = "END"
40  LET I-1(EDITE) = "END"
41  LET I-1(EDITE) = "END"
42  LET I-1(EDITE) = "END"
43  LET I-1(EDITE) = "END"
44  LET I-1(EDITE) = "END"
45  LET I-1(EDITE) = "END"
46  LET I-1(EDITE) = "END"
47  LET I-1(EDITE) = "END"
48  LET I-1(EDITE) = "END"
49  LET I-1(EDITE) = "END"
50  LET I-1(EDITE) = "END"
51  LET I-1(EDITE) = "END"
52  LET I-1(EDITE) = "END"
53  LET I-1(EDITE) = "END"
54  LET I-1(EDITE) = "END"
55  LET I-1(EDITE) = "END"
56  LET I-1(EDITE) = "END"
57  LET I-1(EDITE) = "END"
58  LET I-1(EDITE) = "END"
59  LET I-1(EDITE) = "END"
60  LET I-1(EDITE) = "END"
61  LET I-1(EDITE) = "END"
62  LET I-1(EDITE) = "END"
63  LET I-1(EDITE) = "END"
64  LET I-1(EDITE) = "END"
65  LET I-1(EDITE) = "END"
66  LET I-1(EDITE) = "END"
67  LET I-1(EDITE) = "END"
68  LET I-1(EDITE) = "END"
69  LET I-1(EDITE) = "END"
70  LET I-1(EDITE) = "END"
71  LET I-1(EDITE) = "END"
72  LET I-1(EDITE) = "END"
73  LET I-1(EDITE) = "END"
74  LET I-1(EDITE) = "END"
75  LET I-1(EDITE) = "END"
76  LET I-1(EDITE) = "END"
77  LET I-1(EDITE) = "END"
78  LET I-1(EDITE) = "END"
79  LET I-1(EDITE) = "END"
80  LET I-1(EDITE) = "END"
81  LET I-1(EDITE) = "END"
82  LET I-1(EDITE) = "END"
83  LET I-1(EDITE) = "END"
84  LET I-1(EDITE) = "END"
85  LET I-1(EDITE) = "END"
86  LET I-1(EDITE) = "END"
87  LET I-1(EDITE) = "END"
88  LET I-1(EDITE) = "END"
89  LET I-1(EDITE) = "END"
90  LET I-1(EDITE) = "END"
91  LET I-1(EDITE) = "END"
92  LET I-1(EDITE) = "END"
93  LET I-1(EDITE) = "END"
94  LET I-1(EDITE) = "END"
95  LET I-1(EDITE) = "END"
96  LET I-1(EDITE) = "END"
97  LET I-1(EDITE) = "END"
98  LET I-1(EDITE) = "END"
99  LET I-1(EDITE) = "END"
100 LET I-1(EDITE) = "END"

```

LOCAL VARIETIES OF IRIS CULTIVAR		
IRIS-NEW	1900	7
IRIS-NEW	1900	8
IRIS-NEW	1900	10
IRIS-NEW	1900	11
IRIS-NEW	1900	12
IRIS-NEW	1900	13
IRIS-NEW	1900	14
IRIS-NEW	1900	15
IRIS-NEW	1900	16
IRIS-NEW	1900	17
IRIS-NEW	1900	18
IRIS-NEW	1900	19
IRIS-NEW	1900	20
IRIS-NEW	1900	21
IRIS-NEW	1900	22
IRIS-NEW	1900	23
IRIS-NEW	1900	24
IRIS-NEW	1900	25
IRIS-NEW	1900	26
IRIS-NEW	1900	27
IRIS-NEW	1900	28
IRIS-NEW	1900	29
IRIS-NEW	1900	30
IRIS-NEW	1900	31
IRIS-NEW	1900	32
IRIS-NEW	1900	33
IRIS-NEW	1900	34
IRIS-NEW	1900	35
IRIS-NEW	1900	36
IRIS-NEW	1900	37
IRIS-NEW	1900	38
IRIS-NEW	1900	39
IRIS-NEW	1900	40
IRIS-NEW	1900	41
IRIS-NEW	1900	42
IRIS-NEW	1900	43
IRIS-NEW	1900	44
IRIS-NEW	1900	45
IRIS-NEW	1900	46
IRIS-NEW	1900	47
IRIS-NEW	1900	48
IRIS-NEW	1900	49
IRIS-NEW	1900	50
IRIS-NEW	1900	51
IRIS-NEW	1900	52
IRIS-NEW	1900	53
IRIS-NEW	1900	54
IRIS-NEW	1900	55
IRIS-NEW	1900	56
IRIS-NEW	1900	57
IRIS-NEW	1900	58
IRIS-NEW	1900	59
IRIS-NEW	1900	60
IRIS-NEW	1900	61
IRIS-NEW	1900	62
IRIS-NEW	1900	63
IRIS-NEW	1900	64
IRIS-NEW	1900	65
IRIS-NEW	1900	66
IRIS-NEW	1900	67
IRIS-NEW	1900	68
IRIS-NEW	1900	69
IRIS-NEW	1900	70
IRIS-NEW	1900	71
IRIS-NEW	1900	72
IRIS-NEW	1900	73
IRIS-NEW	1900	74
IRIS-NEW	1900	75
IRIS-NEW	1900	76
IRIS-NEW	1900	77
IRIS-NEW	1900	78
IRIS-NEW	1900	79
IRIS-NEW	1900	80
IRIS-NEW	1900	81
IRIS-NEW	1900	82
IRIS-NEW	1900	83
IRIS-NEW	1900	84
IRIS-NEW	1900	85
IRIS-NEW	1900	86
IRIS-NEW	1900	87
IRIS-NEW	1900	88
IRIS-NEW	1900	89
IRIS-NEW	1900	90
IRIS-NEW	1900	91
IRIS-NEW	1900	92
IRIS-NEW	1900	93
IRIS-NEW	1900	94
IRIS-NEW	1900	95
IRIS-NEW	1900	96
IRIS-NEW	1900	97
IRIS-NEW	1900	98
IRIS-NEW	1900	99
IRIS-NEW	1900	100

55-241

00/01/70 PAGE 45

LINE CACI SIMSCRIPT II.5 RELEASE 44

```

1  EVENT 2.1IF(0.24)
2  DEFINE IPR AS AN INTEGER VARIABLE
3  LET IP = IPR
4  LET 2.1IF(10) = "0"
5  LET IPR(10) = "0"
6  IF IPR(10) = "0"
7  LET I = 1
8  DO
9  IF IP = IPR(0.24)
10  SKIP 1 LINE
11  PRINT 1 LINE
12  PRINT 1 LINE
13  PRINT 1 LINE
14  PRINT 1 LINE
15  PRINT 1 LINE
16  PRINT 1 LINE
17  PRINT 1 LINE
18  PRINT 1 LINE
19  PRINT 1 LINE
20  PRINT 1 LINE
21  PRINT 1 LINE
22  PRINT 1 LINE
23  PRINT 1 LINE
24  PRINT 1 LINE
25  PRINT 1 LINE
26  PRINT 1 LINE
27  PRINT 1 LINE
28  PRINT 1 LINE
29  PRINT 1 LINE
30  PRINT 1 LINE
31  PRINT 1 LINE
32  PRINT 1 LINE
33  PRINT 1 LINE
34  PRINT 1 LINE
35  PRINT 1 LINE
36  PRINT 1 LINE
37  PRINT 1 LINE
38  PRINT 1 LINE
39  PRINT 1 LINE
40  PRINT 1 LINE
41  PRINT 1 LINE
42  PRINT 1 LINE
43  PRINT 1 LINE
44  PRINT 1 LINE
45  PRINT 1 LINE
46  PRINT 1 LINE
47  PRINT 1 LINE
48  PRINT 1 LINE
49  PRINT 1 LINE
50  PRINT 1 LINE
51  PRINT 1 LINE
52  PRINT 1 LINE
53  PRINT 1 LINE
54  PRINT 1 LINE
55  PRINT 1 LINE
56  PRINT 1 LINE
57  PRINT 1 LINE
58  PRINT 1 LINE
59  PRINT 1 LINE
60  PRINT 1 LINE
61  PRINT 1 LINE
62  PRINT 1 LINE
63  PRINT 1 LINE
64  PRINT 1 LINE
65  PRINT 1 LINE
66  PRINT 1 LINE
67  PRINT 1 LINE
68  PRINT 1 LINE
69  PRINT 1 LINE
70  PRINT 1 LINE
71  PRINT 1 LINE
72  PRINT 1 LINE
73  PRINT 1 LINE
74  PRINT 1 LINE
75  PRINT 1 LINE
76  PRINT 1 LINE
77  PRINT 1 LINE
78  PRINT 1 LINE
79  PRINT 1 LINE
80  PRINT 1 LINE
81  PRINT 1 LINE
82  PRINT 1 LINE
83  PRINT 1 LINE
84  PRINT 1 LINE
85  PRINT 1 LINE
86  PRINT 1 LINE
87  PRINT 1 LINE
88  PRINT 1 LINE
89  PRINT 1 LINE
90  PRINT 1 LINE
91  PRINT 1 LINE
92  PRINT 1 LINE
93  PRINT 1 LINE
94  PRINT 1 LINE
95  PRINT 1 LINE
96  PRINT 1 LINE
97  PRINT 1 LINE
98  PRINT 1 LINE
99  PRINT 1 LINE
100 PRINT 1 LINE

```

# LOCAL VARIABLES OF THIS ROUTINE

1.1	INTEGER	WORD 7	1.2	INTEGER	WORD 8
1.3	INTEGER	WORD 9	1.2	INTEGER	WORD 9
1.4	INTEGER	WORD 10	1.3	INTEGER	WORD 10
1.5	INTEGER	WORD 11	1.4	INTEGER	WORD 11
1.6	INTEGER	WORD 12	1.5	INTEGER	WORD 12
1.7	INTEGER	WORD 13	1.6	INTEGER	WORD 13
1.8	INTEGER	WORD 14	1.7	INTEGER	WORD 14
1.9	INTEGER	WORD 15	1.8	INTEGER	WORD 15
1.10	INTEGER	WORD 16	1.9	INTEGER	WORD 16
1.11	INTEGER	WORD 17	1.10	INTEGER	WORD 17
1.12	INTEGER	WORD 18	1.11	INTEGER	WORD 18
1.13	INTEGER	WORD 19	1.12	INTEGER	WORD 19
1.14	INTEGER	WORD 20	1.13	INTEGER	WORD 20
1.15	INTEGER	WORD 21	1.14	INTEGER	WORD 21
1.16	INTEGER	WORD 22	1.15	INTEGER	WORD 22
1.17	INTEGER	WORD 23	1.16	INTEGER	WORD 23
1.18	INTEGER	WORD 24	1.17	INTEGER	WORD 24
1.19	INTEGER	WORD 25	1.18	INTEGER	WORD 25
1.20	INTEGER	WORD 26	1.19	INTEGER	WORD 26
1.21	INTEGER	WORD 27	1.20	INTEGER	WORD 27
1.22	INTEGER	WORD 28	1.21	INTEGER	WORD 28
1.23	INTEGER	WORD 29	1.22	INTEGER	WORD 29
1.24	INTEGER	WORD 30	1.23	INTEGER	WORD 30
1.25	INTEGER	WORD 31	1.24	INTEGER	WORD 31
1.26	INTEGER	WORD 32	1.25	INTEGER	WORD 32
1.27	INTEGER	WORD 33	1.26	INTEGER	WORD 33
1.28	INTEGER	WORD 34	1.27	INTEGER	WORD 34
1.29	INTEGER	WORD 35	1.28	INTEGER	WORD 35
1.30	INTEGER	WORD 36	1.29	INTEGER	WORD 36
1.31	INTEGER	WORD 37	1.30	INTEGER	WORD 37
1.32	INTEGER	WORD 38	1.31	INTEGER	WORD 38
1.33	INTEGER	WORD 39	1.32	INTEGER	WORD 39
1.34	INTEGER	WORD 40	1.33	INTEGER	WORD 40
1.35	INTEGER	WORD 41	1.34	INTEGER	WORD 41
1.36	INTEGER	WORD 42	1.35	INTEGER	WORD 42
1.37	INTEGER	WORD 43	1.36	INTEGER	WORD 43
1.38	INTEGER	WORD 44	1.37	INTEGER	WORD 44
1.39	INTEGER	WORD 45	1.38	INTEGER	WORD 45
1.40	INTEGER	WORD 46	1.39	INTEGER	WORD 46
1.41	INTEGER	WORD 47	1.40	INTEGER	WORD 47
1.42	INTEGER	WORD 48	1.41	INTEGER	WORD 48
1.43	INTEGER	WORD 49	1.42	INTEGER	WORD 49
1.44	INTEGER	WORD 50	1.43	INTEGER	WORD 50
1.45	INTEGER	WORD 51	1.44	INTEGER	WORD 51
1.46	INTEGER	WORD 52	1.45	INTEGER	WORD 52
1.47	INTEGER	WORD 53	1.46	INTEGER	WORD 53
1.48	INTEGER	WORD 54	1.47	INTEGER	WORD 54
1.49	INTEGER	WORD 55	1.48	INTEGER	WORD 55
1.50	INTEGER	WORD 56	1.49	INTEGER	WORD 56
1.51	INTEGER	WORD 57	1.50	INTEGER	WORD 57
1.52	INTEGER	WORD 58	1.51	INTEGER	WORD 58
1.53	INTEGER	WORD 59	1.52	INTEGER	WORD 59
1.54	INTEGER	WORD 60	1.53	INTEGER	WORD 60
1.55	INTEGER	WORD 61	1.54	INTEGER	WORD 61
1.56	INTEGER	WORD 62	1.55	INTEGER	WORD 62
1.57	INTEGER	WORD 63	1.56	INTEGER	WORD 63
1.58	INTEGER	WORD 64	1.57	INTEGER	WORD 64
1.59	INTEGER	WORD 65	1.58	INTEGER	WORD 65
1.60	INTEGER	WORD 66	1.59	INTEGER	WORD 66
1.61	INTEGER	WORD 67	1.60	INTEGER	WORD 67
1.62	INTEGER	WORD 68	1.61	INTEGER	WORD 68
1.63	INTEGER	WORD 69	1.62	INTEGER	WORD 69
1.64	INTEGER	WORD 70	1.63	INTEGER	WORD 70
1.65	INTEGER	WORD 71	1.64	INTEGER	WORD 71
1.66	INTEGER	WORD 72	1.65	INTEGER	WORD 72
1.67	INTEGER	WORD 73	1.66	INTEGER	WORD 73
1.68	INTEGER	WORD 74	1.67	INTEGER	WORD 74
1.69	INTEGER	WORD 75	1.68	INTEGER	WORD 75
1.70	INTEGER	WORD 76	1.69	INTEGER	WORD 76
1.71	INTEGER	WORD 77	1.70	INTEGER	WORD 77
1.72	INTEGER	WORD 78	1.71	INTEGER	WORD 78
1.73	INTEGER	WORD 79	1.72	INTEGER	WORD 79
1.74	INTEGER	WORD 80	1.73	INTEGER	WORD 80
1.75	INTEGER	WORD 81	1.74	INTEGER	WORD 81
1.76	INTEGER	WORD 82	1.75	INTEGER	WORD 82
1.77	INTEGER	WORD 83	1.76	INTEGER	WORD 83
1.78	INTEGER	WORD 84	1.77	INTEGER	WORD 84
1.79	INTEGER	WORD 85	1.78	INTEGER	WORD 85
1.80	INTEGER	WORD 86	1.79	INTEGER	WORD 86
1.81	INTEGER	WORD 87	1.80	INTEGER	WORD 87
1.82	INTEGER	WORD 88	1.81	INTEGER	WORD 88
1.83	INTEGER	WORD 89	1.82	INTEGER	WORD 89
1.84	INTEGER	WORD 90	1.83	INTEGER	WORD 90
1.85	INTEGER	WORD 91	1.84	INTEGER	WORD 91
1.86	INTEGER	WORD 92	1.85	INTEGER	WORD 92
1.87	INTEGER	WORD 93	1.86	INTEGER	WORD 93
1.88	INTEGER	WORD 94	1.87	INTEGER	WORD 94
1.89	INTEGER	WORD 95	1.88	INTEGER	WORD 95
1.90	INTEGER	WORD 96	1.89	INTEGER	WORD 96
1.91	INTEGER	WORD 97	1.90	INTEGER	WORD 97
1.92	INTEGER	WORD 98	1.91	INTEGER	WORD 98
1.93	INTEGER	WORD 99	1.92	INTEGER	WORD 99
1.94	INTEGER	WORD 100	1.93	INTEGER	WORD 100
1.95	INTEGER	WORD 101	1.94	INTEGER	WORD 101
1.96	INTEGER	WORD 102	1.95	INTEGER	WORD 102
1.97	INTEGER	WORD 103	1.96	INTEGER	WORD 103
1.98	INTEGER	WORD 104	1.97	INTEGER	WORD 104
1.99	INTEGER	WORD 105	1.98	INTEGER	WORD 105
1.100	INTEGER	WORD 106	1.99	INTEGER	WORD 106

09/01/76 PAGE 46

```

LINE  CACI SIMSCHIPI 11.5  RELEASE AF
1  EVENT 3.ITE(103E)
2  DEFINE 103E AS AN INTEGER VARIABLE
3  LET 10 = 103E
4  LET 3.ITE(10) = "NO"
5  LET 103(10) = "NO"
6  IF 100 = "YES"
7  DO 1 = 1 17 NOAC(10X)
8  DO
9  IF 10 = 110(10X,1)
10  SKIP 1 OUTPUT LINE
11  PRINT 1 DOUBLE LINE
12  ENW.TIME(AIRPLANF(10)) AS PULLDAS
13  A/C NO. *** WAS INTERNAL THIRD CRACK BECAME EXTERNAL AT *** INCHES AND ***** FLIGHT MOUNHS
14  LEAVE
15  ELSE
16  LOOP
17  ALWAYS
18  RETURN
19  END

```

LOCAL VARIABLES OF THIS ROUTINE

INTEGER	WORD 7	1.2	INTEGER	WORD 8
INTEGER	WORD 9	104E	INTEGER	WORD 1
INTEGER	WORD 10	K.1	INTEGER	WORD 12
INTEGER	WORD 13	K.3	INTEGER	WORD 14
INTEGER	WORD 15	L.4	INTEGER	WORD 3
INTEGER	WORD 17	K.1	DOUBLE	WORD 5

1.1  
1.3  
J.1  
K.2  
K.4  
N.1

09/01/76 PAGE 47

```

LINE CACT SIMSCRIPT II.5 RELEASE MP
1 ROUTINE INSPECTION,SCHEDULE(C)
2 **
3 ** SCHEDULES INSPECTIONS REL. INTERVAL LEVEL ON A GIVEN AIRCRAFT SUCH THAT THE
4 ** AIRCRAFT IS NOT INSPECTED UNTIL IT REACHES MINIMUM DEFECTABLE LGT
5 **
6 OFFICE = AS AN INTERIM VARIABLE
7 LET A1 = ASH(10) * C(10)
8 LET C1 = C.GMRTIM,DATE * C(10)
9 IF FRT,INSP,LEVEL LE 1
10 LET TML = TIME.V + .551/V
11 IF N = 2
12 LET TML = TIME.V + .006/C1
13 ALWAYS
14 LET S,INSP,AT = ENTRY.TIME(AIRPLANE(10)) + 1400(1)
15 * TIME(C.F(TIME-ENTRY.TIME(AIRPLANE(10)))/1400(1)) + 1.0)
16 SCHEDULE AN A,LEVEL,INSPECTION(10) AT S,INSP,AT
17 LET AAL(10) = A,LEVEL,INSPECTION
18 *IF,AND,LESS
19 **
20 IF FRT,INSP,LEVEL LE 2
21 LET TML = TIME.V + .410/M1
22 IF N = 2
23 LET TML = TIME.V + .655/C1
24 ALWAYS
25 LET S,INSP,AT = ENTRY.TIME(AIRPLANE(10)) + 1400(2)
26 * TIME(C.F(TIME-ENTRY.TIME(AIRPLANE(10)))/1400(2)) + 1.0)
27 SCHEDULE A A,LEVEL,INSPECTION(10) AT S,INSP,AT
28 LET AAL(10) = A,LEVEL,INSPECTION
29 *IF,AND,LESS
30 **
31 IF FRT,INSP,LEVEL LE 3
32 LET TML = TIME.V + .200/M1
33 IF N = 2
34 LET TML = TIME.V + .563/C1
35 ALWAYS
36 LET S,INSP,AT = ENTRY.TIME(AIRPLANE(10)) + C,INTERVAL(10)
37 * TIME(C.F(TIME-ENTRY.TIME(AIRPLANE(10)))/C,INTERVAL(10)) + 1.0)
38 SCHEDULE A C,LEVEL,INSPECTION(10) AT S,INSP,AT
39 LET AAL(10) = C,LEVEL,INSPECTION
40 *IF,AND,LESS
41 **
42 LET INSP,SCH(10) = "YES"
43 RETURN
44 END

```

THIS PAGE IS BEST QUALITY PRACTICABLE  
FROM COPY FURNISHED TO DDG

LINE CACJ SIMSCRIPT II.5 RELEASE AF  
LOCAL VARIABLES OF THIS ROUTINE  
09/01/76 PAGE 04  
C1  
N  
TTL  
DOUBLE WORD 5  
INTERSECT WORD 1  
DOUBLE WORD 7  
S-INSPE.A1  
DOUBLE  
WORD 3  
WORD 9

09/11/76 PAGE 19

UNITED STATES DEPARTMENT OF JUSTICE  
FEDERAL BUREAU OF INVESTIGATION  
WASHINGTON, D. C. 20535

[illegible]

4113.2 51141 51154 51167 51180 51193 51206 51219 51232 51245 51258 51271 51284 51297 51310 51323 51336 51349 51362 51375 51388 51401 51414 51427 51440 51453 51466 51479 51492 51505 51518 51531 51544 51557 51570 51583 51596 51609 51622 51635 51648 51661 51674 51687 51700 51713 51726 51739 51752 51765 51778 51791 51804 51817 51830 51843 51856 51869 51882 51895 51908 51921 51934 51947 51960 51973 51986 51999 52012 52025 52038 52051 52064 52077 52090 52103 52116 52129 52142 52155 52168 52181 52194 52207 52220 52233 52246 52259 52272 52285 52298 52311 52324 52337 52350 52363 52376 52389 52402 52415 52428 52441 52454 52467 52480 52493 52506 52519 52532 52545 52558 52571 52584 52597 52610 52623 52636 52649 52662 52675 52688 52701 52714 52727 52740 52753 52766 52779 52792 52805 52818 52831 52844 52857 52870 52883 52896 52909 52922 52935 52948 52961 52974 52987 52999 53012 53025 53038 53051 53064 53077 53090 53103 53116 53129 53142 53155 53168 53181 53194 53207 53220 53233 53246 53259 53272 53285 53298 53311 53324 53337 53350 53363 53376 53389 53402 53415 53428 53441 53454 53467 53480 53493 53506 53519 53532 53545 53558 53571 53584 53597 53610 53623 53636 53649 53662 53675 53688 53701 53714 53727 53740 53753 53766 53779 53792 53805 53818 53831 53844 53857 53870 53883 53896 53909 53922 53935 53948 53961 53974 53987 53999 54012 54025 54038 54051 54064 54077 54090 54103 54116 54129 54142 54155 54168 54181 54194 54207 54220 54233 54246 54259 54272 54285 54298 54311 54324 54337 54350 54363 54376 54389 54402 54415 54428 54441 54454 54467 54480 54493 54506 54519 54532 54545 54558 54571 54584 54597 54610 54623 54636 54649 54662 54675 54688 54701 54714 54727 54740 54753 54766 54779 54792 54805 54818 54831 54844 54857 54870 54883 54896 54909 54922 54935 54948 54961 54974 54987 54999 55012 55025 55038 55051 55064 55077 55090 55103 55116 55129 55142 55155 55168 55181 55194 55207 55220 55233 55246 55259 55272 55285 55298 55311 55324 55337 55350 55363 55376 55389 55402 55415 55428 55441 55454 55467 55480 55493 55506 55519 55532 55545 55558 55571 55584 55597 55610 55623 55636 55649 55662 55675 55688 55701 55714 55727 55740 55753 55766 55779 55792 55805 55818 55831 55844 55857 55870 55883 55896 55909 55922 55935 55948 55961 55974 55987 55999 56012 56025 56038 56051 56064 56077 56090 56103 56116 56129 56142 56155 56168 56181 56194 56207 56220 56233 56246 56259 56272 56285 56298 56311 56324 56337 56350 56363 56376 56389 56402 56415 56428 56441 56454 56467 56480 56493 56506 56519 56532 56545 56558 56571 56584 56597 56610 56623 56636 56649 56662 56675 56688 56701 56714 56727 56740 56753 56766 56779 56792 56805 56818 56831 56844 56857 56870 56883 56896 56909 56922 56935 56948 56961 56974 56987 56999 57012 57025 57038 57051 57064 57077 57090 57103 57116 57129 57142 57155 57168 57181 57194 57207 57220 57233 57246 57259 57272 57285 57298 57311 57324 57337 57350 57363 57376 57389 57402 57415 57428 57441 57454 57467 57480 57493 57506 57519 57532 57545 57558 57571 57584 57597 57610 57623 57636 57649 57662 57675 57688 57701 57714 57727 57740 57753 57766 57779 57792 57805 57818 57831 57844 57857 57870 57883 57896 57909 57922 57935 57948 57961 57974 57987 57999 58012 58025 58038 58051 58064 58077 58090 58103 58116 58129 58142 58155 58168 58181 58194 58207 58220 58233 58246 58259 58272 58285 58298 58311 58324 58337 58350 58363 58376 58389 58402 58415 58428 58441 58454 58467 58480 58493 58506 58519 58532 58545 58558 58571 58584 58597 58610 58623 58636 58649 58662 58675 58688 58701 58714 58727 58740 58753 58766 58779 58792 58805 58818 58831 58844 58857 58870 58883 58896 58909 58922 58935 58948 58961 58974 58987 58999 59012 59025 59038 59051 59064 59077 59090 59103 59116 59129 59142 59155 59168 59181 59194 59207 59220 59233 59246 59259 59272 59285 59298 59311 59324 59337 59350 59363 59376 59389 59402 59415 59428 59441 59454 59467 59480 59493 59506 59519 59532 59545 59558 59571 59584 59597 59610 59623 59636 59649 59662 59675 59688 59701 59714 59727 59740 59753 59766 59779 59792 59805 59818 59831 59844 59857 59870 59883 59896 59909 59922 59935 59948 59961 59974 5

1.  $\text{C}_2\text{H}_5\text{Br}$  +  $\text{H}_2\text{O}$   $\rightarrow$   $\text{C}_2\text{H}_5\text{OH}$  +  $\text{HBr}$



09/01/76 PAGE 50

```

LINE CACI SIMSCRIPT II.5 RELEASE AF
1 EVENT A.LEVEL.INSPECTION(ID)
2 DEFINE IDN AS AN INTEGER VARIABLE
3 LET ID = IDN
4
5 IF EXT.INSPECTION.LEVEL LE 1
6 LET A.LEVEL.INSPECTION = AAL(ID)
7 CANCEL THE A.LEVEL.INSPECTION
8 RESCHEDULE THE A.LEVEL.INSPECTION(ID) AT TIME.V + IACCD(1)
9 LET AAL(ID) = A.LEVEL.INSPECTION
10 ALWAYS
11 LET FIXIT.COST = A.REPAIR.COST
12 LET ASWP = 1.0
13
14 CALL FRAMINF(2,.002,.513,.035,.002,.513,.010) YIELDING FUUND
15
16 SCHEDULE A A.LEVEL.INSPECTION(ID) AT TIME.V + IACCD(2)
17 LET AAL(ID) = A.LEVEL.INSPECTION
18 RETURN
19 END

```

LOCAL VARIABLES OF THIS MODULE

FUUND	NUMBER	TYPE	INTEGER	WORD
	1			

09/01/76 PAGE 51

```

LINE  CACT INSTRUCTIONS 11.0  RECURSE #F
1  GREAT CALCULATIONS(SPECIFIC(LOC))
2  OPERATE FOR AS A: INTERED VARIABLE
3  LET I1 = LOC
4  ..
5  IF I1 = 1 THEN GOTO 12
6  LET I1 = I1 + 1
7  GOTO 11
8  ..
9  ..
10  IF I1 = 1 THEN GOTO 12
11  LET I1 = I1 + 1
12  GOTO 11
13  ..
14  ..
15  IF I1 = 1 THEN GOTO 12
16  LET I1 = I1 + 1
17  GOTO 11
18  ..
19  ..
20  IF I1 = 1 THEN GOTO 12
21  LET I1 = I1 + 1
22  GOTO 11
23  ..
24  ..
25  IF I1 = 1 THEN GOTO 12
26  LET I1 = I1 + 1
27  GOTO 11
28  ..
29  ..
30  IF I1 = 1 THEN GOTO 12
31  LET I1 = I1 + 1
32  GOTO 11
33  ..
34  ..

```

09/01/70 PAGE 52

LINE CACI SCRIPT II.5 RELEASE

```

1  LET I=LEVEL,INSPECTION(100)
2  SET I TO AN INTEGER VARIABLE
3  SET I TO AN INTEGER VARIABLE
4  LET I=100
5  IF I=100 THEN
6    LET I=100
7    DO
8      IF I=100 THEN
9        SET I TO AN INTEGER VARIABLE
10       SET I TO AN INTEGER VARIABLE
11       LET I=100
12       ELSE
13         LOOP
14       ALWAYS
15       IF I=100 THEN
16         LET I=100
17         LET I=100
18         LET I=100
19         LET I=100
20         LET I=100
21         LET I=100
22         LET I=100
23         LET I=100
24         LET I=100
25         LET I=100
26         LET I=100
27         LET I=100
28         LET I=100
29         LET I=100
30         LET I=100
31         LET I=100
32         LET I=100
33         LET I=100
34         LET I=100
35         LET I=100
36         LET I=100
37         LET I=100
38         LET I=100
39         LET I=100
40         LET I=100
41         LET I=100
42         LET I=100
43         LET I=100
44         LET I=100
45         LET I=100
46         LET I=100
47         LET I=100
48         LET I=100
49         LET I=100
50         LET I=100
51         LET I=100
52         LET I=100

```

THIS PAGE IS BEST QUALITY PRACTICABLE  
FROM COPY FURNISHED TO DDC

09/01/76 PAF: 53

```

09/70
LINE CALL SCHEDULE 11.00 RELEASE 44
53 LET AUCO(3) = AUCO(3) + CLEVEL*DECREASE
54 LET AUCO(4) = AUCO(4) + CLEVEL*INCREASE
55 IF LTIME = "YES"
56   GOTO 11.000000 LINE 11
57   1. SCHEDULE 11.000000 AUCO(3), AUCO(4) AS FOLLOWS
      CLEVEL INTERVAL 11.000000 HOURS
      CLEVEL INTERVAL 11.000000 HOURS
58   ALWAYS
59   LET NICHM = NICHM + 1
60   IF LTIME GT 10
61     GOTO 11.000000 LINE 11
62   GOTO 11.000000 LINE 11
63   LET SC(AUCO(3)) = AUCO(3)
64   LET SC(AUCO(4)) = AUCO(4)
65   NEXT
66   LET CLEVEL = AUCO(3)
67   LET NICHM = AUCO(4)
68   GOTO 11.000000 LINE 11
69   GOTO 11.000000 LINE 11
70   LET CLEVEL(TAIL.10) = AUCO(3)
71   LET CLEVEL(TAIL.10) = AUCO(4)
72   GOTO 11.000000 LINE 11
73   GOTO 11.000000 LINE 11
74   GOTO 11.000000 LINE 11
75   LET AUCO(J) = FIVE*V
76   GOTO 11.000000 LINE 11
77   GOTO 11.000000 LINE 11
78   GOTO 11.000000 LINE 11
79   GOTO 11.000000 LINE 11
80   ALWAYS
81   GOTO 11.000000 LINE 11
82   IF TIME*PE*O*IN*O = "YES" OR SPUC*PE*O*IN*O = "YES"
83     SCHEDULE 11.000000 INSPECTION(10) AT TIME*V + D*INTERVAL(10)
84     LET AUCO(10) = CLEVEL*INSPECTION
85     IF TIME*PE*O*IN*O = "YES"
86       GOTO 11.000000 LINE 11
87       GOTO 11.000000 LINE 11
88       IF TIME*V LT FIVE*TIME*PE*O*IN*O + CLEVEL*INTERVAL(10)
89         GOTO 11.000000 LINE 11
90         GOTO 11.000000 LINE 11
91         GOTO 11.000000 LINE 11
92         GOTO 11.000000 LINE 11
93         GOTO 11.000000 LINE 11
94         GOTO 11.000000 LINE 11
95         GOTO 11.000000 LINE 11
96         GOTO 11.000000 LINE 11
97         GOTO 11.000000 LINE 11
98         GOTO 11.000000 LINE 11
99         GOTO 11.000000 LINE 11
100   GOTO 11.000000 LINE 11
101   SCHEDULE 11.000000 LINE 11
102   GOTO 11.000000 LINE 11

```

-173-

[illegible]

09/01/76 PAGE 55

LIST CACI SIMULATOR ILS RELEASE AF

1 ROUTINE FLOWCHART (CASI, CACI, CACI, CACI) YIELDING FLOWCHART

2 OFFER AS AN INTERFER VARIATION

3 OFFER AS AN ALPHA VARIATION

4 IF YES = "YES"

5 END IF = 1 (1) NOAC(LIST)

6

7 IF YES = "YES" THEN

8 LET LIST = 1

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

09/01/76 PAGE 56

LINE CACI SIMSCRIPT II.5 RELEASE HF

```

53 LET TAT = TIME-ACI-STRENGTH-REDUCTION
54 LET CL = (TIME-V-TAT) * 100 + CGRI(10)
55 IF TAT < 100
56 LET CL = (TAT-TAT) * 100 + (TIME-V-TAT) * 100 + CGRI(10)
57 IF CL < 100
58 LET CL = CCL + ((CL-CCL)/100) * 2
59 IF CL < 100
60 IF CL < 100
61 IF CL < 100
62 IF CL < 100
63 IF CL < 100
64 IF CL < 100
65 IF CL < 100
66 IF CL < 100
67 IF CL < 100
68 IF CL < 100
69 IF CL < 100
70 IF CL < 100
71 IF CL < 100
72 IF CL < 100
73 IF CL < 100
74 IF CL < 100
75 IF CL < 100
76 IF CL < 100
77 IF CL < 100
78 IF CL < 100
79 IF CL < 100
80 IF CL < 100
81 IF CL < 100
82 IF CL < 100
83 IF CL < 100
84 IF CL < 100
85 IF CL < 100
86 IF CL < 100
87 IF CL < 100
88 IF CL < 100
89 IF CL < 100
90 IF CL < 100
91 IF CL < 100
92 IF CL < 100
93 IF CL < 100
94 IF CL < 100
95 IF CL < 100
96 IF CL < 100
97 IF CL < 100
98 IF CL < 100
99 IF CL < 100
100 IF CL < 100
101 IF CL < 100
102 IF CL < 100
103 IF CL < 100
104 IF CL < 100

```

66/08/74 WAG 51

44 4574734 7-11 14147415 1247 4-11

```

01  IF N = 1
02  LET ACMAL = CL
03  LET SACOM = CL
04  LET Z = -60
05  JUMP AMT4
06  IF N = 2
07  LET ACMAL = CL
08  LET SACOM = CL
09  LET Z = -60
10  JUMP AMT40
11  IF N = 3
12  LET ACMAL = CL
13  LET SACOM = CL
14  LET Z = -60
15  JUMP AMT40
16  IF N = 4
17  LET ACMAL = CL
18  LET SACOM = CL
19  LET Z = -60
20  JUMP AMT40
21  IF N = 5
22  LET ACMAL = CL
23  LET SACOM = CL
24  LET Z = -60
25  JUMP AMT40
26  IF N = 6
27  LET ACMAL = CL
28  LET SACOM = CL
29  LET Z = -60
30  JUMP AMT40
31  IF N = 7
32  LET ACMAL = CL
33  LET SACOM = CL
34  LET Z = -60
35  JUMP AMT40
36  IF N = 8
37  LET ACMAL = CL
38  LET SACOM = CL
39  LET Z = -60
40  JUMP AMT40
41  IF N = 9
42  LET ACMAL = CL
43  LET SACOM = CL
44  LET Z = -60
45  JUMP AMT40
46  IF N = 10
47  LET ACMAL = CL
48  LET SACOM = CL
49  LET Z = -60
50  JUMP AMT40
51  IF N = 11
52  LET ACMAL = CL
53  LET SACOM = CL
54  LET Z = -60
55  JUMP AMT40
56  IF N = 12
57  LET ACMAL = CL
58  LET SACOM = CL
59  LET Z = -60
60  JUMP AMT40
61  IF N = 13
62  LET ACMAL = CL
63  LET SACOM = CL
64  LET Z = -60
65  JUMP AMT40
66  IF N = 14
67  LET ACMAL = CL
68  LET SACOM = CL
69  LET Z = -60
70  JUMP AMT40
71  IF N = 15
72  LET ACMAL = CL
73  LET SACOM = CL
74  LET Z = -60
75  JUMP AMT40
76  IF N = 16
77  LET ACMAL = CL
78  LET SACOM = CL
79  LET Z = -60
80  JUMP AMT40
81  IF N = 17
82  LET ACMAL = CL
83  LET SACOM = CL
84  LET Z = -60
85  JUMP AMT40
86  IF N = 18
87  LET ACMAL = CL
88  LET SACOM = CL
89  LET Z = -60
90  JUMP AMT40
91  IF N = 19
92  LET ACMAL = CL
93  LET SACOM = CL
94  LET Z = -60
95  JUMP AMT40
96  IF N = 20
97  LET ACMAL = CL
98  LET SACOM = CL
99  LET Z = -60
100 JUMP AMT40

```



THIS PAGE IS BEST QUALITY PRACTICABLE  
FROM COPY FURNISHED TO DDO

[illegible]

AREA	LOCAL	DATE	TIME	STATION	REMARKS
1	177				
CL	177	177	177	177	177
1-3	177	177	177	177	177
4-5	177	177	177	177	177
6-7	177	177	177	177	177
8-9	177	177	177	177	177
10-11	177	177	177	177	177
12-13	177	177	177	177	177
14-15	177	177	177	177	177
16-17	177	177	177	177	177
18-19	177	177	177	177	177
20-21	177	177	177	177	177
22-23	177	177	177	177	177
24-25	177	177	177	177	177
26-27	177	177	177	177	177
28-29	177	177	177	177	177
30-31	177	177	177	177	177
32-33	177	177	177	177	177
34-35	177	177	177	177	177
36-37	177	177	177	177	177
38-39	177	177	177	177	177
40-41	177	177	177	177	177
42-43	177	177	177	177	177
44-45	177	177	177	177	177
46-47	177	177	177	177	177
48-49	177	177	177	177	177
50-51	177	177	177	177	177
52-53	177	177	177	177	177
54-55	177	177	177	177	177
56-57	177	177	177	177	177
58-59	177	177	177	177	177
60-61	177	177	177	177	177
62-63	177	177	177	177	177
64-65	177	177	177	177	177
66-67	177	177	177	177	177
68-69	177	177	177	177	177
70-71	177	177	177	177	177
72-73	177	177	177	177	177
74-75	177	177	177	177	177
76-77	177	177	177	177	177
78-79	177	177	177	177	177
80-81	177	177	177	177	177
82-83	177	177	177	177	177
84-85	177	177	177	177	177
86-87	177	177	177	177	177
88-89	177	177	177	177	177
90-91	177	177	177	177	177
92-93	177	177	177	177	177
94-95	177	177	177	177	177
96-97	177	177	177	177	177
98-99	177	177	177	177	177
100-101	177	177	177	177	177
102-103	177	177	177	177	177
104-105	177	177	177	177	177
106-107	177	177	177	177	177
108-109	177	177	177	177	177
110-111	177	177	177	177	177
112-113	177	177	177	177	177
114-115	177	177	177	177	177
116-117	177	177	177	177	177
118-119	177	177	177	177	177
120-121	177	177	177	177	177
122-123	177	177	177	177	177
124-125	177	177	177	177	177

08/01/76 PAGE 50

LINE CASE SIMULATION ILLS RELEASE 44

1. MONITORING MODIFICATIONS

2. COMPUTES RESPONSIBILITY OF DETECTING A SERVICE DEFECT OF A GIVEN SIZE
3. COMPUTES RESPONSIBILITY OF DETECTING A LINE DEFECT OF A GIVEN SIZE
4. COMPUTES RESPONSIBILITY OF DETECTING A CONSTRUCTION DEFECT OF A GIVEN SIZE
5. COMPUTES RESPONSIBILITY OF DETECTING A DEFECT OF A GIVEN SIZE
6. COMPUTES RESPONSIBILITY OF DETECTING A DEFECT OF A GIVEN SIZE
7. COMPUTES RESPONSIBILITY OF DETECTING A DEFECT OF A GIVEN SIZE
8. COMPUTES RESPONSIBILITY OF DETECTING A DEFECT OF A GIVEN SIZE
9. COMPUTES RESPONSIBILITY OF DETECTING A DEFECT OF A GIVEN SIZE
10. COMPUTES RESPONSIBILITY OF DETECTING A DEFECT OF A GIVEN SIZE
11. COMPUTES RESPONSIBILITY OF DETECTING A DEFECT OF A GIVEN SIZE
12. COMPUTES RESPONSIBILITY OF DETECTING A DEFECT OF A GIVEN SIZE

13. COMPUTES RESPONSIBILITY OF DETECTING A DEFECT OF A GIVEN SIZE

LOCAL VARIATIONS OF THIS METHOD

14. COMPUTES RESPONSIBILITY OF DETECTING A DEFECT OF A GIVEN SIZE

-179-

-179-

09/01/70 PAGE 01

```

LINE  CECI INSTRUCTION  (L.S.  RELEASE  RE
1  EVENT REACH-FALL-SAFE-LOGIC(ONES)
2  DEFINE LOGS AS AN INTEGER VARIABLE
3  LET IO = LOGS
4  IF LOGS = "YES"
5  THEN LET IO = LOGS
6  ELSE LET IO = LOGS(102)
7  IF IO = LOGS(102)
8  THEN LET IO = LOGS(102)
9  LET IO = LOGS(102)
10  LET IO = LOGS(102)
11  LET IO = LOGS(102)
12  LET IO = LOGS(102)
13  LET IO = LOGS(102)
14  LET IO = LOGS(102)
15  LET IO = LOGS(102)
16  LET IO = LOGS(102)
17  LET IO = LOGS(102)
18  LET IO = LOGS(102)
19  LET IO = LOGS(102)
20  LET IO = LOGS(102)
21  LET IO = LOGS(102)
22  LET IO = LOGS(102)
23  LET IO = LOGS(102)
24  LET IO = LOGS(102)
25  LET IO = LOGS(102)
26  LET IO = LOGS(102)
27  LET IO = LOGS(102)
28  LET IO = LOGS(102)

```

LOCAL VARIABLES OF THIS ROUTINE

1.1	LOGS	WORD 7	1.2	LOGS	WORD 4
1.3	LOGS	WORD 8	1.4	LOGS	WORD 1
1.5	LOGS	WORD 9	1.6	LOGS	WORD 12
1.7	LOGS	WORD 10	1.8	LOGS	WORD 14
1.9	LOGS	WORD 11	1.10	LOGS	WORD 15
1.11	LOGS	WORD 12	1.12	LOGS	WORD 16
1.13	LOGS	WORD 13	1.14	LOGS	WORD 17
1.15	LOGS	WORD 14	1.16	LOGS	WORD 18
1.17	LOGS	WORD 15	1.18	LOGS	WORD 19
1.19	LOGS	WORD 16	1.20	LOGS	WORD 20
1.21	LOGS	WORD 17	1.22	LOGS	WORD 21
1.23	LOGS	WORD 18	1.24	LOGS	WORD 22
1.25	LOGS	WORD 19	1.26	LOGS	WORD 23
1.27	LOGS	WORD 20	1.28	LOGS	WORD 24
1.29	LOGS	WORD 21	1.30	LOGS	WORD 25
1.31	LOGS	WORD 22	1.32	LOGS	WORD 26
1.33	LOGS	WORD 23	1.34	LOGS	WORD 27
1.35	LOGS	WORD 24	1.36	LOGS	WORD 28
1.37	LOGS	WORD 25	1.38	LOGS	WORD 29
1.39	LOGS	WORD 26	1.40	LOGS	WORD 30
1.41	LOGS	WORD 27	1.42	LOGS	WORD 31
1.43	LOGS	WORD 28	1.44	LOGS	WORD 32
1.45	LOGS	WORD 29	1.46	LOGS	WORD 33
1.47	LOGS	WORD 30	1.48	LOGS	WORD 34
1.49	LOGS	WORD 31	1.50	LOGS	WORD 35
1.51	LOGS	WORD 32	1.52	LOGS	WORD 36
1.53	LOGS	WORD 33	1.54	LOGS	WORD 37
1.55	LOGS	WORD 34	1.56	LOGS	WORD 38
1.57	LOGS	WORD 35	1.58	LOGS	WORD 39
1.59	LOGS	WORD 36	1.60	LOGS	WORD 40
1.61	LOGS	WORD 37	1.62	LOGS	WORD 41
1.63	LOGS	WORD 38	1.64	LOGS	WORD 42
1.65	LOGS	WORD 39	1.66	LOGS	WORD 43
1.67	LOGS	WORD 40	1.68	LOGS	WORD 44
1.69	LOGS	WORD 41	1.70	LOGS	WORD 45
1.71	LOGS	WORD 42	1.72	LOGS	WORD 46
1.73	LOGS	WORD 43	1.74	LOGS	WORD 47
1.75	LOGS	WORD 44	1.76	LOGS	WORD 48
1.77	LOGS	WORD 45	1.78	LOGS	WORD 49
1.79	LOGS	WORD 46	1.80	LOGS	WORD 50
1.81	LOGS	WORD 47	1.82	LOGS	WORD 51
1.83	LOGS	WORD 48	1.84	LOGS	WORD 52
1.85	LOGS	WORD 49	1.86	LOGS	WORD 53
1.87	LOGS	WORD 50	1.88	LOGS	WORD 54
1.89	LOGS	WORD 51	1.90	LOGS	WORD 55
1.91	LOGS	WORD 52	1.92	LOGS	WORD 56
1.93	LOGS	WORD 53	1.94	LOGS	WORD 57
1.95	LOGS	WORD 54	1.96	LOGS	WORD 58
1.97	LOGS	WORD 55	1.98	LOGS	WORD 59
1.99	LOGS	WORD 56	1.100	LOGS	WORD 60

09/01/76 PAGE 62

LINE CACI SYNSCRIPT 11.5 RELEASE RF

```

1 EVENT FAILURE(IUFA)
2 DEFINE IOFA AS AN INTEGER VARIABLE
3 DEFINE MOLD AS AN INTEGER VARIABLE
4 LET IO = IOFA
5 IF LTM = "YES"
6   FOR I = 1 TO NOAC(10X)
7     DO
8       IF IO = TLIN(LDX,1)
9         SKIP 1 LINE WITH LINE
10        PRINT 1 LINE WITH ID, TIME,V-ENTRY,TIME(AIRPLANE(ID)) AS FOLLOWS
11        A/C NO. AND EXPERIENCES (LEFT-FAILURE AT ***** FLIGHT HOURS
12        LET 1.STRINGM.REDUCTION = A1SW(ID)
13        LET 1A1 = TIME.A(1.STRINGM.REDUCTION)
14        LET CCL = CRIT.CRA.LGT
15        IF C1.EXISTS(ID) = "YES"
16          LET COMSTON = AC(ID)
17          LET 1AC = TIME.A(COMSTON)
18          ALWAYS
19          LET 1CL = (TIME.V-TA1)*SW(ID)*CGWI(ID)
20          IF 1AC GT 1A1
21            LET 1CL = (1AC-TA1)*MSR(ID) + (TIME.V-TAC)*MSR(ID)*CGWI(ID)
22            ALWAYS
23            IF 1CL GT CCL
24              LET 1CL = CCL + (1CL-CCL)*(MSW(ID)/MSW(ID))
25            ALWAYS
26            IF 2-CR.EXISTS(ID) = "YES"
27              LET 2.STRINGM.REDUCTION = A2SR(ID)
28              LET 2A2 = TIME.A(2.STRINGM.REDUCTION)
29              LET 2CL = (TIME.V-TA2)*SR(ID)*CGWI(ID)
30              IF 2AC GT 2A2
31                LET 2CL = (2AC-TA2)*MSR(ID) + (TIME.V-TAC)*MSR(ID)*CGWI(ID)
32                ALWAYS
33                IF 2CL GT CCL
34                  LET 2CL = CCL + (2CL-CCL)*(MSW(ID)/MSW(ID))
35                ALWAYS
36                IF 3-CR.EXISTS(ID) = "YES"
37                  LET 3.STRINGM.REDUCTION = A3SW(ID)
38                  LET 3A3 = TIME.A(3.STRINGM.REDUCTION)
39                  LET 3CL = (TIME.V-TA3)*SW(ID)*CGWI(ID)
40                  IF 3AC GT 3A3
41                    LET 3CL = (3AC-TA3)*SW(ID) + (TIME.V-TAC)*SW(ID)*CGWI(ID)
42                    ALWAYS
43                    IF 3CL GT CCL
44                      LET 3CL = CCL + (3CL-CCL)*(MSW(ID)/MSW(ID))
45                      ALWAYS
46                      ALWAYS
47                      LET CL = 1CL+2CL+3CL
48                      LET S = 2.75 - CL*(1.75/ESAF.LGT)
49                      IF CL GT ESAF.LGT
50                        LET S = (LGM,IN).FAILURE-CL/(LGMT,IN).FAILURE-ESAF.LGT
51                      ALWAYS
52                      PRINT 2 LINES WITH CL, S/2.75 AS FOLLOWS

```

09/01/76 PAGE 03

LINE 0001 SIMSCRIPT ILS RELEASE OF

SIZE OF CRACK LENGTHS AT FAILURE = \*\*\* INCHES  
RESIDUAL STRENGTH AT FAILURE = \*\* ULTIMATE

LEAVE

ELSE

LOOP

ALWAYS

SCHEMATIC AN INCREASE INSPECTION FREQUENCY AND

SCHEMATIC AN IMMEDIATE FUEL INSPECTION RUN

LET FSH(10) = 0.00

IF SPSL(1) = 0

LET SPSL = SPSL + 1

LET A(10)(USPL) = 10

LET FLW(USPL) = TIME.V - ENTRY.TIME(AIRPLANE(10))

IF FLW(USPL) = 0

IF SPSL(1) = 100

LET SPSL = SPSL + 1

LET SPSL(SPSL) = 10

LET SPSL(SPSL) = TIME.V - ENTRY.TIME(AIRPLANE(10))

LET SPSL(SPSL) = FLW(USPL)

IF FLW(USPL) = 0

FOR I = 1 TO 10

DO

IF MI.TIME.ACFT(I) = 10

ADD 1 TO LMTA

IF LMTA(1) = 10

IF LMTA(1) = 10

IF LMTA(1) = 10

IF LMTA(1) = 10

IF LMTA(1) = 10

IF LMTA(1) = 10

IF LMTA(1) = 10

IF LMTA(1) = 10

IF LMTA(1) = 10

IF LMTA(1) = 10

IF LMTA(1) = 10

IF LMTA(1) = 10

IF LMTA(1) = 10

IF LMTA(1) = 10

IF LMTA(1) = 10

IF LMTA(1) = 10

IF LMTA(1) = 10

IF LMTA(1) = 10

IF LMTA(1) = 10

IF LMTA(1) = 10

IF LMTA(1) = 10

IF LMTA(1) = 10

IF LMTA(1) = 10

IF LMTA(1) = 10

IF LMTA(1) = 10

IF LMTA(1) = 10

IF LMTA(1) = 10

IF LMTA(1) = 10

IF LMTA(1) = 10

IF LMTA(1) = 10

IF LMTA(1) = 10

IF LMTA(1) = 10

IF LMTA(1) = 10

IF LMTA(1) = 10

IF LMTA(1) = 10

IF LMTA(1) = 10

IF LMTA(1) = 10

IF LMTA(1) = 10

IF LMTA(1) = 10

IF LMTA(1) = 10

IF LMTA(1) = 10

IF LMTA(1) = 10

09/01/76 PAGE 64

```

104 LINE CACI SIMSCRIPT II.5 WELFARE AF
105 REGARDLESS
106 REMOVE AIRPLANE(10) FROM ACTIVE.FLEET
107 FILE AIRPLANE(10) IN CRASHED.FLEET
108 ADD 1 TO 2.NUM.OF.CRASH
109 ** CANCEL SCHEDULED EVENTS AND DESTROY EVENT NOTICES
110 **
111 IF 1E(10) = "YES"
112 LET 1.LIFE = 1E(10)
113 CANCEL THE 1.LIFE
114 DESTROY THE 1.LIFE
115 IF 1E(10) = "YES"
116 LET 2.LIFE = 2E(10)
117 CANCEL THE 2.LIFE
118 DESTROY THE 2.LIFE
119 IF 1E(10) = "YES"
120 LET 3.LIFE = 3E(10)
121 CANCEL THE 3.LIFE
122 DESTROY THE 3.LIFE
123 ALWAYS
124 ALWAYS
125 ALWAYS
126 IF COL.EXISTS(10) = "N"
127 LET COMBUSTION = AC(10)
128 IF COL.EXISTS(10) = "N"
129 CANCEL THE COMBUSTION
130 ALWAYS
131 DESTROY THE COMBUSTION
132 LET COL.EXISTS(10) = "N"
133 **ENDLESS
134 IF 1.CM.EXISTS(10) = "N"
135 LET 1.STRENGTH.REDUCTION = 1E(10)
136 IF 1.CM.EXISTS(10) = "N"
137 CANCEL THE 1.STRENGTH.REDUCTION
138 ALWAYS
139 DESTROY THE 1.STRENGTH.REDUCTION
140 LET 1.CM.EXISTS(10) = "N"
141 IF 2.CM.EXISTS(10) = "N"
142 LET 2.STRENGTH.REDUCTION = 2E(10)
143 IF 2.CM.EXISTS(10) = "N"
144 CANCEL THE 2.STRENGTH.REDUCTION
145 ALWAYS
146 DESTROY THE 2.STRENGTH.REDUCTION
147 LET 2.CM.EXISTS(10) = "N"
148 IF 3.CM.EXISTS(10) = "N"
149 LET 3.STRENGTH.REDUCTION = 3E(10)
150 IF 3.CM.EXISTS(10) = "N"
151 CANCEL THE 3.STRENGTH.REDUCTION
152 ALWAYS
153 DESTROY THE 3.STRENGTH.REDUCTION
154 LET 3.CM.EXISTS(10) = "N"
155 REGARDLESS
156 REGARDLESS

```

09/01/76 PAGE 85

LINE CACI ST-SCH-INT T1.5 RELEASE AT

```

157 *END*
158 IF ALL(0) = "YES"
159 LET WEACH.FAIL.SAF.LGT = AMFSL(0)
160 CANCEL THE WEACH.FAIL.SAF.LGT
161 DESTROY THE WEACH.FAIL.SAF.LGT
162 IF ALL(0) = "NO"
163 *END*
164 LET D.LEVEL.INSPECTION = AMFSL(0)
165 CANCEL THE D.LEVEL.INSPECTION
166 DESTROY THE D.LEVEL.INSPECTION
167 IF D.LEVEL.INSPECTION = "YES"
168 CALL CANCEL.SCHEDULE.INSPECTIONS
169 *END*
170 **
171 FOR EACH WEACH.FAIL.SAF.LGT IN EV.SCHEDULE.INSPECTIONS
172 DO
173 IF TIME = 1) CANCEL THE WEACH.FAIL.SAF.LGT
174 DESTROY THE WEACH.FAIL.SAF.LGT
175 LEAVE
176 ELSE
177 FOR EACH T1.SERVICE.DAMAGE IN EV.SCHEDULE.INSPECTIONS
178 DO
179 IF TIME = 1) CANCEL THE T1.SERVICE.DAMAGE
180 DESTROY THE T1.SERVICE.DAMAGE
181 LEAVE
182 ELSE
183 FOR EACH T1.INSPECTION.INCREASE IN EV.SCHEDULE.INSPECTIONS
184 DO
185 IF TIME = 1) CANCEL THE T1.INSPECTION.INCREASE
186 DESTROY THE T1.INSPECTION.INCREASE
187 LEAVE
188 ELSE
189 RETURN
190 *END*

```

LOCAL VARIABLES OF THIS ROUTINE

COL	WORD 19	CL	WORD 33
COL	WORD 3	CL	WORD 7
COL	WORD 8	CL	WORD 9
COL	WORD 1	CL	WORD 10
COL	WORD 12	CL	WORD 15
COL	WORD 14	CL	WORD 18
COL	WORD 17	CL	WORD 21
COL	WORD 20	CL	WORD 24
COL	WORD 23	CL	WORD 27
COL	WORD 26	CL	WORD 30
COL	WORD 29	CL	WORD 33
COL	WORD 32	CL	WORD 36
COL	WORD 35	CL	WORD 39
COL	WORD 38	CL	WORD 42
COL	WORD 41	CL	WORD 45
COL	WORD 44	CL	WORD 48
COL	WORD 47	CL	WORD 51
COL	WORD 50	CL	WORD 54
COL	WORD 53	CL	WORD 57
COL	WORD 56	CL	WORD 60
COL	WORD 59	CL	WORD 63
COL	WORD 62	CL	WORD 66
COL	WORD 65	CL	WORD 69
COL	WORD 68	CL	WORD 72
COL	WORD 71	CL	WORD 75
COL	WORD 74	CL	WORD 78
COL	WORD 77	CL	WORD 81
COL	WORD 80	CL	WORD 84
COL	WORD 83	CL	WORD 87
COL	WORD 86	CL	WORD 90
COL	WORD 89	CL	WORD 93
COL	WORD 92	CL	WORD 96
COL	WORD 95	CL	WORD 99
COL	WORD 98	CL	WORD 102
COL	WORD 101	CL	WORD 105
COL	WORD 104	CL	WORD 108
COL	WORD 103	CL	WORD 111
COL	WORD 110	CL	WORD 114
COL	WORD 109	CL	WORD 117
COL	WORD 108	CL	WORD 120
COL	WORD 107	CL	WORD 123
COL	WORD 106	CL	WORD 126
COL	WORD 105	CL	WORD 129
COL	WORD 104	CL	WORD 132
COL	WORD 103	CL	WORD 135
COL	WORD 102	CL	WORD 138
COL	WORD 101	CL	WORD 141
COL	WORD 100	CL	WORD 144
COL	WORD 99	CL	WORD 147
COL	WORD 98	CL	WORD 150
COL	WORD 97	CL	WORD 153
COL	WORD 96	CL	WORD 156
COL	WORD 95	CL	WORD 159
COL	WORD 94	CL	WORD 162
COL	WORD 93	CL	WORD 165
COL	WORD 92	CL	WORD 168
COL	WORD 91	CL	WORD 171
COL	WORD 90	CL	WORD 174
COL	WORD 89	CL	WORD 177
COL	WORD 88	CL	WORD 180
COL	WORD 87	CL	WORD 183
COL	WORD 86	CL	WORD 186
COL	WORD 85	CL	WORD 189
COL	WORD 84	CL	WORD 192
COL	WORD 83	CL	WORD 195
COL	WORD 82	CL	WORD 198
COL	WORD 81	CL	WORD 201
COL	WORD 80	CL	WORD 204
COL	WORD 79	CL	WORD 207
COL	WORD 78	CL	WORD 210
COL	WORD 77	CL	WORD 213
COL	WORD 76	CL	WORD 216
COL	WORD 75	CL	WORD 219
COL	WORD 74	CL	WORD 222
COL	WORD 73	CL	WORD 225
COL	WORD 72	CL	WORD 228
COL	WORD 71	CL	WORD 231
COL	WORD 70	CL	WORD 234
COL	WORD 69	CL	WORD 237
COL	WORD 68	CL	WORD 240
COL	WORD 67	CL	WORD 243
COL	WORD 66	CL	WORD 246
COL	WORD 65	CL	WORD 249
COL	WORD 64	CL	WORD 252
COL	WORD 63	CL	WORD 255
COL	WORD 62	CL	WORD 258
COL	WORD 61	CL	WORD 261
COL	WORD 60	CL	WORD 264
COL	WORD 59	CL	WORD 267
COL	WORD 58	CL	WORD 270
COL	WORD 57	CL	WORD 273
COL	WORD 56	CL	WORD 276
COL	WORD 55	CL	WORD 279
COL	WORD 54	CL	WORD 282
COL	WORD 53	CL	WORD 285
COL	WORD 52	CL	WORD 288
COL	WORD 51	CL	WORD 291
COL	WORD 50	CL	WORD 294
COL	WORD 49	CL	WORD 297
COL	WORD 48	CL	WORD 300
COL	WORD 47	CL	WORD 303
COL	WORD 46	CL	WORD 306
COL	WORD 45	CL	WORD 309
COL	WORD 44	CL	WORD 312
COL	WORD 43	CL	WORD 315
COL	WORD 42	CL	WORD 318
COL	WORD 41	CL	WORD 321
COL	WORD 40	CL	WORD 324
COL	WORD 39	CL	WORD 327
COL	WORD 38	CL	WORD 330
COL	WORD 37	CL	WORD 333
COL	WORD 36	CL	WORD 336
COL	WORD 35	CL	WORD 339
COL	WORD 34	CL	WORD 342
COL	WORD 33	CL	WORD 345
COL	WORD 32	CL	WORD 348
COL	WORD 31	CL	WORD 351
COL	WORD 30	CL	WORD 354
COL	WORD 29	CL	WORD 357
COL	WORD 28	CL	WORD 360
COL	WORD 27	CL	WORD 363
COL	WORD 26	CL	WORD 366
COL	WORD 25	CL	WORD 369
COL	WORD 24	CL	WORD 372
COL	WORD 23	CL	WORD 375
COL	WORD 22	CL	WORD 378
COL	WORD 21	CL	WORD 381
COL	WORD 20	CL	WORD 384
COL	WORD 19	CL	WORD 387
COL	WORD 18	CL	WORD 390
COL	WORD 17	CL	WORD 393
COL	WORD 16	CL	WORD 396
COL	WORD 15	CL	WORD 399
COL	WORD 14	CL	WORD 402
COL	WORD 13	CL	WORD 405
COL	WORD 12	CL	WORD 408
COL	WORD 11	CL	WORD 411
COL	WORD 10	CL	WORD 414
COL	WORD 9	CL	WORD 417
COL	WORD 8	CL	WORD 420
COL	WORD 7	CL	WORD 423
COL	WORD 6	CL	WORD 426
COL	WORD 5	CL	WORD 429
COL	WORD 4	CL	WORD 432
COL	WORD 3	CL	WORD 435
COL	WORD 2	CL	WORD 438
COL	WORD 1	CL	WORD 441
COL	WORD 0	CL	WORD 444



THIS PAGE IS BEST QUALITY PRACTICABLE  
FROM COPY FURNISHED TO DDC

```

LINE CALL SIMSCNOUT 11.5 RELEASE NO
1 EVENT AT TIME, FLOW, SERVICE (INDEL)
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
```

- 186 -

09/01/76 PAGE 68

LINE CACI SIMSCRIPT II.5 RELEASE AF

```

106 MEGAROLESS
107 LET D.LEVEL.INSPECTION = ADL(ID)
108 CANCEL THE D.LEVEL.INSPECTION
109 DESTROY THE D.LEVEL.INSPECTION
110 IF ATL(ID) = "YES"
111 LET REACH.FAIL.SAFE.LGT = AWFSL(ID)
112 CANCEL THE REACH.FAIL.SAFE.LGT
113 DESTROY THE REACH.FAIL.SAFE.LGT
114 LET ALL(ID) = "NO"
115 REGRADLESS
116 IF FSW(ID) = "YES"
117 LET FAILURE = AF(ID)
118 CANCEL THE FAILURE
119 DESTROY THE FAILURE
120 LET FSW(ID) = "NO"
121 ALWAYS
122 "
123 RETURN
124 END

```

LOCAL VARIABLES OF THIS ROUTINE

ADL	ADL	3	I.1	INTEGER	WORD 7
AF	AF	4	I.3	INTEGER	WORD 9
ATL	ATL	1	J.1	INTEGER	WORD 10
AWFSL	AWFSL	12	K.2	INTEGER	WORD 15
AWFSL	AWFSL	18	K.3	INTEGER	WORD 15
AWFSL	AWFSL	10	L.2	INTEGER	WORD 9
AWFSL	AWFSL	11	M.1	DOUBLE	WORD 5

09/01/76 PAGE 69

```

L14 CALL STASCP111.5 RELEASE AF
1 EVENT REMAIN(IMP)
2 DEFINE TIMEP AS AN INTEGER VARIABLE
3 LET ID = IDREP
4 IF TIMEPENDING(ID) = "YES" OR SMUD.PENDING(ID) = "YES"
5 IF TIMEPENDING(ID) = "YES"
6 CALL INSTALLCOMPICATION
7 LET TIMEPENDING(ID) = "NO"
8 IF TIMEPENDING(ID) = "NO"
9 LET TIMEPENDING(ID) = "NO"
10 LET TIMEPENDING(ID) = "NO"
11 LET TIMEPENDING(ID) = "NO"
12 LET TIMEPENDING(ID) = "NO"
13 LET TIMEPENDING(ID) = "NO"
14 LET TIMEPENDING(ID) = "NO"
15 LET TIMEPENDING(ID) = "NO"
16 LET TIMEPENDING(ID) = "NO"
17 LET TIMEPENDING(ID) = "NO"
18 LET TIMEPENDING(ID) = "NO"
19 LET TIMEPENDING(ID) = "NO"
20 LET TIMEPENDING(ID) = "NO"
21 LET TIMEPENDING(ID) = "NO"
22 LET TIMEPENDING(ID) = "NO"
23 LET TIMEPENDING(ID) = "NO"
24 LET TIMEPENDING(ID) = "NO"
25 LET TIMEPENDING(ID) = "NO"
26 LET TIMEPENDING(ID) = "NO"
27 LET TIMEPENDING(ID) = "NO"
28 LET TIMEPENDING(ID) = "NO"
29 LET TIMEPENDING(ID) = "NO"
30 LET TIMEPENDING(ID) = "NO"
31 LET TIMEPENDING(ID) = "NO"
32 LET TIMEPENDING(ID) = "NO"
33 LET TIMEPENDING(ID) = "NO"
34 LET TIMEPENDING(ID) = "NO"
35 LET TIMEPENDING(ID) = "NO"
36 LET TIMEPENDING(ID) = "NO"
37 LET TIMEPENDING(ID) = "NO"
38 LET TIMEPENDING(ID) = "NO"
39 LET TIMEPENDING(ID) = "NO"
40 LET TIMEPENDING(ID) = "NO"
41 LET TIMEPENDING(ID) = "NO"
42 LET TIMEPENDING(ID) = "NO"
43 LET TIMEPENDING(ID) = "NO"
44 LET TIMEPENDING(ID) = "NO"
45 LET TIMEPENDING(ID) = "NO"
46 LET TIMEPENDING(ID) = "NO"
47 LET TIMEPENDING(ID) = "NO"
48 LET TIMEPENDING(ID) = "NO"
49 LET TIMEPENDING(ID) = "NO"
50 LET TIMEPENDING(ID) = "NO"
51 LET TIMEPENDING(ID) = "NO"
52 LET TIMEPENDING(ID) = "NO"
53 LET TIMEPENDING(ID) = "NO"

```

09/01/76 PAGE 70

LINE CACI SIMSCRIPT II.5 HELFASE AF

```

52 LET CL = (TIME.V-TA1) * MSR(ID) * CGM1(ID)
53 IF TAC GT TA1
54 LET CL = (TAC-TA1)*MSR(ID) + (TIME.V-TAC)*MSR(ID)*CGM1(ID)
55 MEGAROLFS
56 IF CCL LT CL
57 LET CL = CCL + ((CCL-CCL)/MSR(ID))*MSR(ID)
58 MEGAROLFS
59 LET MAX.CRK = CL
60 LET SIM.WED = CL
61 IF 2.CM.FAITS(ID) = "YES"
62 LET 2.SIM.WED = 2.SIM.WED
63 LET 2.SIM.WED = 2.SIM.WED
64 LET TA2 = TIME.A1(2.SIM.WED.REDUCTION)
65 LET CL = (TIME.V-TA2) * MSR(ID) * CGM1(ID)
66 IF TAC GT TA2
67 LET CL = (TAC-TA2)*MSR(ID) + (TIME.V-TAC)*MSR(ID)*CGM1(ID)
68 MEGAROLFS
69 IF CCL LT CL
70 LET CL = CCL + ((CCL-CCL)/MSR(ID))*MSR(ID)
71 MEGAROLFS
72 IF MAX.CRK LT CL
73 LET MAX.CRK = CL
74 MEGAROLFS
75 ADD CL TO STR.WED
76 IF 3.CM.EXISTS(ID) = "YES"
77 LET 3.SIM.WED = 3.SIM.WED
78 LET TA3 = TIME.A1(3.SIM.WED.REDUCTION)
79 LET CL = (TIME.V-TA3) * MSR(ID) * CGM1(ID)
80 IF TAC GT TA3
81 LET CL = (TAC-TA3)*MSR(ID) + (TIME.V-TAC)*MSR(ID)*CGM1(ID)
82 MEGAROLFS
83 IF CCL LT CL
84 LET CL = CCL + ((CCL-CCL)/MSR(ID))*MSR(ID)
85 MEGAROLFS
86 IF MAX.CRK LT CL
87 LET MAX.CRK = CL
88 MEGAROLFS
89 ADD CL TO STR.WED
90 MEGAROLFS
91 LET 4.SIM.WED = 4.SIM.WED
92 LET 4.SIM.WED = 4.SIM.WED
93 LET 4.SIM.WED = 4.SIM.WED
94 LET 4.SIM.WED = 4.SIM.WED
95 LET 4.SIM.WED = 4.SIM.WED
96 LET 4.SIM.WED = 4.SIM.WED
97 LET 4.SIM.WED = 4.SIM.WED
98 LET 4.SIM.WED = 4.SIM.WED
99 LET 4.SIM.WED = 4.SIM.WED
100 LET 4.SIM.WED = 4.SIM.WED
101 LET 4.SIM.WED = 4.SIM.WED
102 LET 4.SIM.WED = 4.SIM.WED
103 LET 4.SIM.WED = 4.SIM.WED
104 LET 4.SIM.WED = 4.SIM.WED
105 LET 4.SIM.WED = 4.SIM.WED
106 LET 4.SIM.WED = 4.SIM.WED
107 LET 4.SIM.WED = 4.SIM.WED
108 LET 4.SIM.WED = 4.SIM.WED
109 LET 4.SIM.WED = 4.SIM.WED
110 LET 4.SIM.WED = 4.SIM.WED
111 LET 4.SIM.WED = 4.SIM.WED
112 LET 4.SIM.WED = 4.SIM.WED
113 LET 4.SIM.WED = 4.SIM.WED
114 LET 4.SIM.WED = 4.SIM.WED
115 LET 4.SIM.WED = 4.SIM.WED
116 LET 4.SIM.WED = 4.SIM.WED
117 LET 4.SIM.WED = 4.SIM.WED
118 LET 4.SIM.WED = 4.SIM.WED
119 LET 4.SIM.WED = 4.SIM.WED
120 LET 4.SIM.WED = 4.SIM.WED

```

09/01/76 PAGE 71

LT-4 CACI STASCH-1 11.5 RELEASE RE

```

107 IF FLEET,STR,RED GT (PSAF,LCI,5,0) * (INDM-1,NUM,OF,RETIME-2,NUM,OF,CRASH)
108 SCHEDULE AN INCREASE,INSPECTION,FREQUENCY NDA
109 SCHEDULE AN IMMEDIATE,FLEET,INSPECTION NDA
110 LET TIME,OF,RETIME = TIME,V
111 REMARK LESS
112 IF-
113 **
114 LET COST,OF,REPAIRS = CUST,OF,REPAIRS + FLXIT,COST
115 IF TIME,OF,RETIME = "N" AND DEC,OF,RETIME = "NO"
116 IF IS,FAILURE = "YES" AND TIME,V LT START,TEST + ACTUAL,AVG,PAT,LIFE /
117 TEST,ACCEL,FACT OR IS,FAILURE = "YES" AND TIME,V GT START,TEST +
118 ACTUAL,AVG,PAT,LIFE / TEST,ACCEL,FACT + LPAQ,TIME OR IS,FAILURE = "NO"
119 IF TIME,OF,RETIME = "YES" AND TIME,V LT START,TEST + ACTUAL,AVG,PAT,LIFE /
120 TEST,ACCEL,FACT OR IS,FAILURE = "YES" AND TIME,V GT START,TEST +
121 ACTUAL,AVG,PAT,LIFE / TEST,ACCEL,FACT + LPAQ,TIME OR IS,FAILURE = "NO"
122 SCHEDULE A REVISION,OF,RETIME NDA
123 LET RE,OF,RETIME = "YES"
124 REMARK LESS
125 REMARK LESS
126 REMARK LESS
127 **
128 LET AAFL = ACTUAL,AVG,PAT,LIFE
129 IF I LE FOUR
130 LET AAFL = 1AAFL
131 REMARK LESS
132 CALL PAT,OF,LIFE,SCATTER(AAFL,5) YIELDING FIRST,LIFE, SECOND,LIFE AND
133 THIRD,LIFE
134 LET 1,STRENGTH,REDUCTION = ASK(10)
135 LET 2,STRENGTH,REDUCTION = ASK(10)
136 LET 3,STRENGTH,REDUCTION = ASK(10)
137 IF 2,STRENGTH,REDUCTION = "YES"
138 DESTROY THE 2,STRENGTH,REDUCTION
139 IF 3,STRENGTH,REDUCTION = "YES"
140 DESTROY THE 3,STRENGTH,REDUCTION
141 IF 1,STRENGTH,REDUCTION = "YES"
142 IF 1,STRENGTH,REDUCTION = "YES"
143 IF 1,STRENGTH,REDUCTION = "YES"
144 IF 1,STRENGTH,REDUCTION = "YES"
145 IF 1,STRENGTH,REDUCTION = "YES"
146 IF 1,STRENGTH,REDUCTION = "YES"
147 IF 1,STRENGTH,REDUCTION = "YES"
148 IF 1,STRENGTH,REDUCTION = "YES"
149 IF 1,STRENGTH,REDUCTION = "YES"
150 IF 1,STRENGTH,REDUCTION = "YES"
151 IF 1,STRENGTH,REDUCTION = "YES"
152 IF 1,STRENGTH,REDUCTION = "YES"
153 IF 1,STRENGTH,REDUCTION = "YES"
154 IF 1,STRENGTH,REDUCTION = "YES"
155 IF 1,STRENGTH,REDUCTION = "YES"
156 IF 1,STRENGTH,REDUCTION = "YES"
157 IF 1,STRENGTH,REDUCTION = "YES"
158 IF 1,STRENGTH,REDUCTION = "YES"
159 IF 1,STRENGTH,REDUCTION = "YES"
160 IF 1,STRENGTH,REDUCTION = "YES"

```

THIS PAGE IS BEST QUALITY PRACTICABLE  
FROM COPY FURNISHED TO DDG

09/01/76 PAGE 72

LINE CACI SCRIPT II.5 RELEASE 8F

```

160 DESTROY THE 3.STRENGTH.REDUCTION
161 JUMP AHEAD
162 OTHERWISE
163 LET THIRD.LIFE = USAGE.LIFE
164 JUMP AHEAD
165 OTHERWISE
166 IF 2.CR.EXISTS(10) = "NS"
167 IF 3.CR.EXISTS(10) = "NS"
168 IF TIME.A(3.STRENGTH.REDUCTION) - TIME.V LT FIRST.LIFE
169 LET THIRD.LIFE = FIRST.LIFE
170 LET SECOND.LIFE = TIME.A(3.STRENGTH.REDUCTION) - TIME.V
171 LET FIRST.LIFE = TIME.A(2.STRENGTH.REDUCTION) - TIME.V
172 CANCEL THE 2.STRENGTH.REDUCTION
173 DESTROY THE 2.STRENGTH.REDUCTION
174 CANCEL THE 3.STRENGTH.REDUCTION
175 DESTROY THE 3.STRENGTH.REDUCTION
176 JUMP AHEAD
177 OTHERWISE
178 IF TIME.A(2.STRENGTH.REDUCTION) - TIME.V LT FIRST.LIFE
179 LET THIRD.LIFE = TIME.A(3.STRENGTH.REDUCTION) - TIME.V
180 LET SECOND.LIFE = FIRST.LIFE
181 LET FIRST.LIFE = TIME.A(2.STRENGTH.REDUCTION) - TIME.V
182 CANCEL THE 2.STRENGTH.REDUCTION
183 DESTROY THE 2.STRENGTH.REDUCTION
184 CANCEL THE 3.STRENGTH.REDUCTION
185 DESTROY THE 3.STRENGTH.REDUCTION
186 JUMP AHEAD
187 OTHERWISE
188 LET SECOND.LIFE = TIME.A(2.STRENGTH.REDUCTION)
189 LET THIRD.LIFE = TIME.A(3.STRENGTH.REDUCTION)
190 CANCEL THE 2.STRENGTH.REDUCTION
191 DESTROY THE 2.STRENGTH.REDUCTION
192 CANCEL THE 3.STRENGTH.REDUCTION
193 DESTROY THE 3.STRENGTH.REDUCTION
194 JUMP AHEAD
195 OTHERWISE
196 IF TIME.A(2.STRENGTH.REDUCTION) - TIME.V LT FIRST.LIFE
197 LET THIRD.LIFE = USAGE.LIFE
198 LET SECOND.LIFE = FIRST.LIFE
199 LET FIRST.LIFE = TIME.A(2.STRENGTH.REDUCTION)
200 CANCEL THE 2.STRENGTH.REDUCTION
201 DESTROY THE 2.STRENGTH.REDUCTION
202 JUMP AHEAD
203 OTHERWISE
204 LET SECOND.LIFE = TIME.A(2.STRENGTH.REDUCTION)
205 JUMP AHEAD
206 OTHERWISE
207 LET THIRD.LIFE = USAGE.LIFE
208 LET FIRST.LIFE = USAGE.LIFE
209 HERE
210 LET 1.CR.EXISTS(10) = "NN"
211 LET 2.CR.EXISTS(10) = "NN"
212 LET 3.CR.EXISTS(10) = "NN"

```

09/01/76 PAGE 73

LINE (ACI) SCRIPT IL-5 RELEASE AF

```

213 IF FIRSTLIFE LT 1ST OR COLLEXTS(10) = "NS" OR SO.SCH(10) = "YES"
214   SCHEDULE A 1-STEP GTH.REDUCTION(10) AT TIME.V + FIRSTLIFE
215   LET A1SCH(10) = 1-STEP GTH.REDUCTION
216   LET 1-STEP GTH.REDUCTION(10) = "NS"
217 IF SECONDALIFE LT 1ST OR COLLEXTS(10) = "NS" OR SO.SCH(10) = "YES"
218   SCHEDULE A 2-STEP GTH.REDUCTION(10) AT TIME.V + SECONDALIFE
219   LET A2SCH(10) = 2-STEP GTH.REDUCTION
220   LET 2-STEP GTH.REDUCTION(10) = "NS"
221 IF THIRDLIFE LT 1ST OR COLLEXTS(10) = "NS" OR SO.SCH(10) = "YES"
222   SCHEDULE A 3-STEP GTH.REDUCTION(10) AT TIME.V + THIRDLIFE
223   LET A3SCH(10) = 3-STEP GTH.REDUCTION
224   LET 3-STEP GTH.REDUCTION(10) = "NS"
225   REMARKLESS
226   REMARKLESS
227   REMARKLESS
228   REMARKLESS
229   IF 1P(10) = "YES"
230     LET 1-STEP GTH.REDUCTION(10) = "NS"
231     LET 1P(10) = "NO"
232   CANCEL THE 1-STEP
233   DESTROY THE 1-STEP
234   IF 1P(10) = "YES"
235     LET 2-STEP GTH.REDUCTION(10) = "NS"
236     LET 1P(10) = "NO"
237   CANCEL THE 2-STEP
238   DESTROY THE 2-STEP
239   IF 1P(10) = "YES"
240     LET 3-STEP GTH.REDUCTION(10) = "NS"
241     LET 1P(10) = "NO"
242   CANCEL THE 3-STEP
243   DESTROY THE 3-STEP
244   ALWAYS
245   ALWAYS
246   ALWAYS
247   IF LIME = "YES"
248     FOR I = 1 TO 100 (LIME)
249       IF I = 100 (LIME)
250         SKIP 1 OUTPUT LINE
251         PRINT A LIME WITH ID, TIME.V-5000, TIME(10-PLANE(10)), FIRSTLIFE,
252           SECONDALIFE, THIRDLIFE AS FOLLOWS
253           A/C 100 WAS ALL DEFECTS REPAIRED AT ***** FLIGHT HOURS
254           1ST CHECK INITIATION PROJECTED AT ***** FLIGHT HOURS
255           2ND CHECK INITIATION PROJECTED AT ***** FLIGHT HOURS
256           3RD CHECK INITIATION PROJECTED AT ***** FLIGHT HOURS
257       LEAVE
258     FLSH
259     LOOP
260     ALWAYS
261   ** CANCEL SCHEDULED INSPECTIONS ON THIS AIRPORT
262   **
263   CALL CANCEL.SCHEDULED.INSPECTIONS

```



THIS PAGE IS BEST QUALITY PRACTICABLE  
FROM COPY FURNISHED TO DDC

09/01/76 PAGE 74

LINE CACI SCRIPT II.5 RELEASE 44

202 RETURN  
203 END

LOCAL VARIABLES IN THIS ROUTINE

AAFL	DOUBLE	WORD 25	CCL	DOUBLE	WORD 9
CL	DOUBLE	WORD 13	FIRST.LIFE	DOUBLE	WORD 27
WORDS.TU.C	DOUBLE	WORD 7	I.1	INTEGER	WORD 37
I.2	INTEGER	WORD 39	I.3	INTEGER	WORD 39
INFP	INTEGER	WORD 1	I.1	INTEGER	WORD 40
K.1	INTEGER	WORD 42	K.2	INTEGER	WORD 43
K.3	INTEGER	WORD 44	K.4	INTEGER	WORD 45
L.4	INTEGER	WORD 35	AAFL.CK	DOUBLE	WORD 15
M.1	INTEGER	WORD 41	OTI.CK	DOUBLE	WORD 23
SPC IND.LIF	DOUBLE	WORD 29	AST	DOUBLE	WORD 17
TAC	DOUBLE	WORD 5	ST.460	DOUBLE	WORD 11
T82	DOUBLE	WORD 19	T81	DOUBLE	WORD 21
THIRD.LIFE	DOUBLE	WORD 31	T83	DOUBLE	WORD 21

THIS PAGE IS BEST QUALITY PRACTICABLE  
FROM COPY FURNISHED TO DDC

09/01/76 PAGE 75

```

11 5 CACTI SYNOPSIS II.5 RELEASE ME
1  EVALUATE INCREMENTAL INCREASE (IOTI)
2  DEFINE IOTI AS AN INTEGER VARIABLE
3  LET IOTI = IOTI
4  IF IOTI = 0 THEN LET IOTI = 1
5  LET IOTI = IOTI + 1
6  LET IOTI = IOTI + 1
7  LET IOTI = IOTI + 1
8  LET IOTI = IOTI + 1
9  LET IOTI = IOTI + 1
10 LET IOTI = IOTI + 1
11 LET IOTI = IOTI + 1
12 LET IOTI = IOTI + 1
13 LET IOTI = IOTI + 1
14 LET IOTI = IOTI + 1
15 LET IOTI = IOTI + 1
16 LET IOTI = IOTI + 1
17 LET IOTI = IOTI + 1
18 LET IOTI = IOTI + 1
19 LET IOTI = IOTI + 1
20 LET IOTI = IOTI + 1
21 LET IOTI = IOTI + 1
22 LET IOTI = IOTI + 1
23 LET IOTI = IOTI + 1
24 LET IOTI = IOTI + 1
25 LET IOTI = IOTI + 1
26 LET IOTI = IOTI + 1
27 LET IOTI = IOTI + 1
28 LET IOTI = IOTI + 1
29 LET IOTI = IOTI + 1
30 LET IOTI = IOTI + 1
31 LET IOTI = IOTI + 1
32 LET IOTI = IOTI + 1
33 LET IOTI = IOTI + 1
34 LET IOTI = IOTI + 1
35 LET IOTI = IOTI + 1
36 LET IOTI = IOTI + 1
37 LET IOTI = IOTI + 1
38 LET IOTI = IOTI + 1
39 LET IOTI = IOTI + 1
40 LET IOTI = IOTI + 1
41 LET IOTI = IOTI + 1
42 LET IOTI = IOTI + 1
43 LET IOTI = IOTI + 1
44 LET IOTI = IOTI + 1
45 LET IOTI = IOTI + 1
46 LET IOTI = IOTI + 1
47 LET IOTI = IOTI + 1
48 LET IOTI = IOTI + 1
49 LET IOTI = IOTI + 1
50 LET IOTI = IOTI + 1
51 LET IOTI = IOTI + 1
52 LET IOTI = IOTI + 1
53 LET IOTI = IOTI + 1
54 LET IOTI = IOTI + 1
55 LET IOTI = IOTI + 1
56 LET IOTI = IOTI + 1
57 LET IOTI = IOTI + 1
58 LET IOTI = IOTI + 1
59 LET IOTI = IOTI + 1
60 LET IOTI = IOTI + 1
61 LET IOTI = IOTI + 1
62 LET IOTI = IOTI + 1
63 LET IOTI = IOTI + 1
64 LET IOTI = IOTI + 1
65 LET IOTI = IOTI + 1
66 LET IOTI = IOTI + 1
67 LET IOTI = IOTI + 1
68 LET IOTI = IOTI + 1
69 LET IOTI = IOTI + 1
70 LET IOTI = IOTI + 1
71 LET IOTI = IOTI + 1
72 LET IOTI = IOTI + 1
73 LET IOTI = IOTI + 1
74 LET IOTI = IOTI + 1
75 LET IOTI = IOTI + 1
76 LET IOTI = IOTI + 1
77 LET IOTI = IOTI + 1
78 LET IOTI = IOTI + 1
79 LET IOTI = IOTI + 1
80 LET IOTI = IOTI + 1
81 LET IOTI = IOTI + 1
82 LET IOTI = IOTI + 1
83 LET IOTI = IOTI + 1
84 LET IOTI = IOTI + 1
85 LET IOTI = IOTI + 1
86 LET IOTI = IOTI + 1
87 LET IOTI = IOTI + 1
88 LET IOTI = IOTI + 1
89 LET IOTI = IOTI + 1
90 LET IOTI = IOTI + 1
91 LET IOTI = IOTI + 1
92 LET IOTI = IOTI + 1
93 LET IOTI = IOTI + 1
94 LET IOTI = IOTI + 1
95 LET IOTI = IOTI + 1
96 LET IOTI = IOTI + 1
97 LET IOTI = IOTI + 1
98 LET IOTI = IOTI + 1
99 LET IOTI = IOTI + 1
100 LET IOTI = IOTI + 1

```

LOCAL VARIABLES IN THIS ROUTINE

IOTI = 1

IOTI

09/01/76 PAGE 76

LIVE CACI SCRIPT II.5 RELEASE AF

```

1  EVFMT INCREASE,INSPECTION,FREQUENCY
2  **
3  ** INCREASES THE FREQUENCY OF THE LOWEST LEVEL INTERNAL AND EXTERNAL INSPECTIONS
4  ** THE SP MAY BE MAY NOT BE THE SAME LEVEL OF INSPECTION
5  ** THE A-LEVEL AND N-LEVEL FREQUENCIES ARE NEVER CHANGED
6  **
7  LET FILET,ST,RED = 0.0
8  IF ARCD(LIL)*S.FREQ.CMG GE IARCD(LIL-1) OR ARCD(LEL)*S.FREQ.CMG GE IARCD(LEL-1)
9  IF ARCD(LIL)*S.FREQ.CMG GE IARCD(LIL-1)
10 LET DIFF = ARCD(LIL) - ARCD(LIL-1) * S.FREQ.CMG
11 LET ARCD(LIL) = ARCD(LIL) + S.FREQ.CMG
12 ALWAYS
13 IF LIL NE LEL
14 IF ARCD(LEL)*S.FREQ.CMG GE IARCD(LEL-1)
15 LET ARCD(LEL) = ARCD(LEL-1) + S.FREQ.CMG
16 ALWAYS
17 IF LTIMO = "YES"
18 SETP 1,INITIAL LINE
19 5 LINES WITH ARCD(3), ARCD(4) AS FOLLOWS
20 INSPECTION INTERVAL DECREASE IMPLEMENTED
    C-LEVEL INTERVAL WITH ***** HOURS
    D-LEVEL INTERVAL WITH ***** MINUTES
    ALWAYS
21 LET CINSL = ARCD(3)
22 LET DINSL = ARCD(4)
23 LET NINSL = ARCD(4)
24 LET NINSL = NINSL + 1
25 IF NINSL GT 14
26 JUMP AHEAD
27 OTHERWISE
28 LET SC(NINSL) = ARCD(3)
29 LET SC(NINSL) = ARCD(4)
30 MEMF
31 FROM EVENT AIRCRAFT IN ACTIVE,FILET
32 ON
33 LET IN = TAIL.IN
34 LET C.INTERVAL(IN) = ARCD(3)
35 LET D.INTERVAL(IN) = ARCD(4)
36 IF LIL = 4
37 LET N.LEVEL.INSPECTION = ARCD(IN)
38 CANCEL THE N.LEVEL.INSPECTION
39 IF TIME-ARCD.LEVEL.INSPECTION - TIME.V LT DIFX
40 RESCHEDULE THE D.LEVEL.INSPECTION(IN) AND
    JUMP AHEAD
41 OTHERWISE
42 RESCHEDULE THE D.LEVEL.INSPECTION(IN) AT TIME-ARCD.LEVEL.INSPECTION - DIFF
43 JUMP AHEAD
44 OTHERWISE
45 JUMP AHEAD
46 OTHERWISE
47 IF INSP_SC(IN) = "YES"
48 LET C.LEVEL.INSPECTION = ARCD(IN)
49 CANCEL THE C.LEVEL.INSPECTION
50 IF TIME-ARCD.LEVEL.INSPECTION - TIME.V LT DIFF

```

THIS PAGE IS BEST QUALITY PRACTICABLE  
FROM COPY FURNISHED TO DDC

92/01/76  
part  
77

[illegible]

LOCAL	VARIABLES OF	THIS QUALITY	NAME	NUMBER
QUALITY	NAME	1-1	INTEREST	NAME 7
INTEREST	NAME	1-2	INTEREST	NAME 8
INTEREST	NAME	1-3	INTEREST	NAME 9
INTEREST	NAME	1-4	INTEREST	NAME 10
INTEREST	NAME	1-5	INTEREST	NAME 11
INTEREST	NAME	1-6	INTEREST	NAME 12
INTEREST	NAME	1-7	INTEREST	NAME 13
INTEREST	NAME	1-8	INTEREST	NAME 14
INTEREST	NAME	1-9	INTEREST	NAME 15
INTEREST	NAME	1-10	INTEREST	NAME 16
INTEREST	NAME	1-11	INTEREST	NAME 17
INTEREST	NAME	1-12	INTEREST	NAME 18
INTEREST	NAME	1-13	INTEREST	NAME 19
INTEREST	NAME	1-14	INTEREST	NAME 20
INTEREST	NAME	1-15	INTEREST	NAME 21
INTEREST	NAME	1-16	INTEREST	NAME 22
INTEREST	NAME	1-17	INTEREST	NAME 23
INTEREST	NAME	1-18	INTEREST	NAME 24
INTEREST	NAME	1-19	INTEREST	NAME 25
INTEREST	NAME	1-20	INTEREST	NAME 26
INTEREST	NAME	1-21	INTEREST	NAME 27
INTEREST	NAME	1-22	INTEREST	NAME 28
INTEREST	NAME	1-23	INTEREST	NAME 29
INTEREST	NAME	1-24	INTEREST	NAME 30
INTEREST	NAME	1-25	INTEREST	NAME 31
INTEREST	NAME	1-26	INTEREST	NAME 32
INTEREST	NAME	1-27	INTEREST	NAME 33
INTEREST	NAME	1-28	INTEREST	NAME 34
INTEREST	NAME	1-29	INTEREST	NAME 35
INTEREST	NAME	1-30	INTEREST	NAME 36
INTEREST	NAME	1-31	INTEREST	NAME 37
INTEREST	NAME	1-32	INTEREST	NAME 38
INTEREST	NAME	1-33	INTEREST	NAME 39
INTEREST	NAME	1-34	INTEREST	NAME 40
INTEREST	NAME	1-35	INTEREST	NAME 41
INTEREST	NAME	1-36	INTEREST	NAME 42
INTEREST	NAME	1-37	INTEREST	NAME 43
INTEREST	NAME	1-38	INTEREST	NAME 44
INTEREST	NAME	1-39	INTEREST	NAME 45
INTEREST	NAME	1-40	INTEREST	NAME 46
INTEREST	NAME	1-41	INTEREST	NAME 47
INTEREST	NAME	1-42	INTEREST	NAME 48
INTEREST	NAME	1-43	INTEREST	NAME 49
INTEREST	NAME	1-44	INTEREST	NAME 50
INTEREST	NAME	1-45	INTEREST	NAME 51
INTEREST	NAME	1-46	INTEREST	NAME 52
INTEREST	NAME	1-47	INTEREST	NAME 53
INTEREST	NAME	1-48	INTEREST	NAME 54
INTEREST	NAME	1-49	INTEREST	NAME 55
INTEREST	NAME	1-50	INTEREST	NAME 56
INTEREST	NAME	1-51	INTEREST	NAME 57
INTEREST	NAME	1-52	INTEREST	NAME 58
INTEREST	NAME	1-53	INTEREST	NAME 59
INTEREST	NAME	1-54	INTEREST	NAME 60
INTEREST	NAME	1-55	INTEREST	NAME 61
INTEREST	NAME	1-56	INTEREST	NAME 62
INTEREST	NAME	1-57	INTEREST	NAME 63
INTEREST	NAME	1-58	INTEREST	NAME 64
INTEREST	NAME	1-59	INTEREST	NAME 65
INTEREST	NAME	1-60	INTEREST	NAME 66
INTEREST	NAME	1-61	INTEREST	NAME 67
INTEREST	NAME	1-62	INTEREST	NAME 68
INTEREST	NAME	1-63	INTEREST	NAME 69
INTEREST	NAME	1-64	INTEREST	NAME 70
INTEREST	NAME	1-65	INTEREST	NAME 71
INTEREST	NAME	1-66	INTEREST	NAME 72
INTEREST	NAME	1-67	INTEREST	NAME 73
INTEREST	NAME	1-68	INTEREST	NAME 74
INTEREST	NAME	1-69	INTEREST	NAME 75
INTEREST	NAME	1-70	INTEREST	NAME 76
INTEREST	NAME	1-71	INTEREST	NAME 77
INTEREST	NAME	1-72	INTEREST	NAME 78
INTEREST	NAME	1-73	INTEREST	NAME 79
INTEREST	NAME	1-74	INTEREST	NAME 80
INTEREST	NAME	1-75	INTEREST	NAME 81
INTEREST	NAME	1-76	INTEREST	NAME 82
INTEREST	NAME	1-77	INTEREST	NAME 83
INTEREST	NAME	1-78	INTEREST	NAME 84
INTEREST	NAME	1-79	INTEREST	NAME 85
INTEREST	NAME	1-80	INTEREST	NAME 86
INTEREST	NAME	1-81	INTEREST	NAME 87

09/01/76 PAGE 7A

```

LINE CACI SCHEDULED INSPECTION RELEASE AF
1 EVENT IMMEDIATE FLEET INSPECTION
2 **
3 ** THIS EVENT REPRESENTS AN IMMEDIATE FLEET WIDE INSPECTION CAUSED BY FINDING A
4 ** DEFECT CONSIDERED TO BE/ADVERSE TO DEPEND ON SCHEDULED INSPECTIONS
5 **
6 ** THIS EVENT IS ALWAYS ENTERED BY THE FIRST INCREASE INSPECTION FREQUENCY
7 **
8 IF LTMS = "YES"
9   SETP 1 ENTER IT LINE
10   GOINT 1 LINE AS FOLLOWS
11     FLEET WIDE SPECIAL INSPECTION PERFORMED
12     ALWAYS
13     LET PCL = CALL COMBAT
14     LET MSIC = ASIC * 1
15     LET FIRSTACT = SCHEDULED COST
16     LET TODAY ATTEMPT TO ACTIVATE FLEET
17     LET LIST = 0.0
18     LET I = 1
19     IF LTMS = "YES"
20       GOINT 1 LINE AS FOLLOWS
21       LET I = 1
22       LET LIST = 0.0
23       LET I = 1
24       LET LIST = 1.0
25       LET I = 1
26       LET I = 1
27       LET I = 1
28       LET I = 1
29       LET I = 1
30       LET I = 1
31       LET I = 1
32       LET I = 1
33       LET I = 1
34       LET I = 1
35       LET I = 1
36       LET I = 1
37       LET I = 1
38       LET I = 1
39       LET I = 1
40       LET I = 1
41       LET I = 1
42       LET I = 1
43       LET I = 1
44       LET I = 1
45       LET I = 1
46       LET I = 1
47       LET I = 1
48       LET I = 1
49       LET I = 1
50       LET I = 1
51       LET I = 1
52       LET I = 1
53       LET I = 1
54       LET I = 1
55       LET I = 1
56       LET I = 1
57       LET I = 1
58       LET I = 1
59       LET I = 1
60       LET I = 1
61       LET I = 1
62       LET I = 1
63       LET I = 1
64       LET I = 1
65       LET I = 1
66       LET I = 1
67       LET I = 1
68       LET I = 1
69       LET I = 1
70       LET I = 1
71       LET I = 1
72       LET I = 1
73       LET I = 1
74       LET I = 1
75       LET I = 1
76       LET I = 1
77       LET I = 1
78       LET I = 1
79       LET I = 1
80       LET I = 1
81       LET I = 1
82       LET I = 1
83       LET I = 1
84       LET I = 1
85       LET I = 1
86       LET I = 1
87       LET I = 1
88       LET I = 1
89       LET I = 1
90       LET I = 1
91       LET I = 1
92       LET I = 1
93       LET I = 1
94       LET I = 1
95       LET I = 1
96       LET I = 1
97       LET I = 1
98       LET I = 1
99       LET I = 1
100      LET I = 1

```

... AT ... FLIGHT ...

09/01/76 PAGE 79

LINE CACI SIMSCRIPT ILS RELEASE AF

```

52 LET SIMSCRIPT_RELEASED = AISC(10)
53 LET TAT = TIME-ARC-SIMSCRIPT_RELEASED
54 LET CL = (TIME-V-TAT) * 10 * (CGL(10))
55 IF TAT < 100
56 LET CL = (SIMSCRIPT_RELEASED * (TIME-V-TAT) * 10 * (CGL(10)))
57 + CGL(10)
58 IF CL < 100
59 LET CL = CL * (CGL(10) * 10)
60 IF CL < 100
61 LET CL = CL * 10
62 LET CL = CL * 10
63 LET CL = CL * 10
64 LET CL = CL * 10
65 LET CL = CL * 10
66 LET CL = CL * 10
67 LET CL = CL * 10
68 LET CL = CL * 10
69 LET CL = CL * 10
70 LET CL = CL * 10
71 LET CL = CL * 10
72 LET CL = CL * 10
73 LET CL = CL * 10
74 LET CL = CL * 10
75 LET CL = CL * 10
76 LET CL = CL * 10
77 LET CL = CL * 10
78 LET CL = CL * 10
79 LET CL = CL * 10
80 LET CL = CL * 10
81 LET CL = CL * 10
82 LET CL = CL * 10
83 LET CL = CL * 10
84 LET CL = CL * 10
85 LET CL = CL * 10
86 LET CL = CL * 10
87 LET CL = CL * 10
88 LET CL = CL * 10
89 LET CL = CL * 10
90 LET CL = CL * 10
91 LET CL = CL * 10
92 LET CL = CL * 10
93 LET CL = CL * 10
94 LET CL = CL * 10
95 LET CL = CL * 10
96 LET CL = CL * 10
97 LET CL = CL * 10
98 LET CL = CL * 10
99 LET CL = CL * 10
100 LET CL = CL * 10
101 LET CL = CL * 10
102 LET CL = CL * 10

```

10/01/75 PAGE 20

LIN CACI SUGGESTION 11.5 RELEASE AF

```

103 **
104 **
105 **
106 **
107 **
108 **
109 **
110 **
111 **
112 **
113 **
114 **
115 **
116 **
117 **
118 **
119 **
120 **
121 **
122 **
123 **
124 **
125 **
126 **
127 **
128 **
129 **
130 **
131 **
132 **
133 **
134 **
135 **
136 **
137 **
138 **
139 **
140 **
141 **
142 **
143 **
144 **
145 **
146 **
147 **
148 **
149 **
150 **
151 **
152 **
153 **
154 **
155 **
156 **
157 **
158 **
159 **
160 **
161 **
162 **
163 **
164 **
165 **
166 **
167 **
168 **
169 **
170 **
171 **
172 **
173 **
174 **
175 **
176 **
177 **
178 **
179 **
180 **
181 **
182 **
183 **
184 **
185 **
186 **
187 **
188 **
189 **
190 **
191 **
192 **
193 **
194 **
195 **
196 **
197 **
198 **
199 **
200 **
201 **
202 **
203 **
204 **
205 **
206 **
207 **
208 **
209 **
210 **
211 **
212 **
213 **
214 **
215 **
216 **
217 **
218 **
219 **
220 **
221 **
222 **
223 **
224 **
225 **
226 **
227 **
228 **
229 **
230 **
231 **
232 **
233 **
234 **
235 **
236 **
237 **
238 **
239 **
240 **
241 **
242 **
243 **
244 **
245 **
246 **
247 **
248 **
249 **
250 **
251 **
252 **
253 **
254 **
255 **
256 **
257 **
258 **
259 **
260 **
261 **
262 **
263 **
264 **
265 **
266 **
267 **
268 **
269 **
270 **
271 **
272 **
273 **
274 **
275 **
276 **
277 **
278 **
279 **
280 **
281 **
282 **
283 **
284 **
285 **
286 **
287 **
288 **
289 **
290 **
291 **
292 **
293 **
294 **
295 **
296 **
297 **
298 **
299 **
300 **
301 **
302 **
303 **
304 **
305 **
306 **
307 **
308 **
309 **
310 **
311 **
312 **
313 **
314 **
315 **
316 **
317 **
318 **
319 **
320 **
321 **
322 **
323 **
324 **
325 **
326 **
327 **
328 **
329 **
330 **
331 **
332 **
333 **
334 **
335 **
336 **
337 **
338 **
339 **
340 **
341 **
342 **
343 **
344 **
345 **
346 **
347 **
348 **
349 **
350 **
351 **
352 **
353 **
354 **
355 **
356 **
357 **
358 **
359 **
360 **
361 **
362 **
363 **
364 **
365 **
366 **
367 **
368 **
369 **
370 **
371 **
372 **
373 **
374 **
375 **
376 **
377 **
378 **
379 **
380 **
381 **
382 **
383 **
384 **
385 **
386 **
387 **
388 **
389 **
390 **
391 **
392 **
393 **
394 **
395 **
396 **
397 **
398 **
399 **
400 **
401 **
402 **
403 **
404 **
405 **
406 **
407 **
408 **
409 **
410 **
411 **
412 **
413 **
414 **
415 **
416 **
417 **
418 **
419 **
420 **
421 **
422 **
423 **
424 **
425 **
426 **
427 **
428 **
429 **
430 **
431 **
432 **
433 **
434 **
435 **
436 **
437 **
438 **
439 **
440 **
441 **
442 **
443 **
444 **
445 **
446 **
447 **
448 **
449 **
450 **
451 **
452 **
453 **
454 **
455 **
456 **
457 **
458 **
459 **
460 **
461 **
462 **
463 **
464 **
465 **
466 **
467 **
468 **
469 **
470 **
471 **
472 **
473 **
474 **
475 **
476 **
477 **
478 **
479 **
480 **
481 **
482 **
483 **
484 **
485 **
486 **
487 **
488 **
489 **
490 **
491 **
492 **
493 **
494 **
495 **
496 **
497 **
498 **
499 **
500 **
501 **
502 **
503 **
504 **
505 **
506 **
507 **
508 **
509 **
510 **
511 **
512 **
513 **
514 **
515 **
516 **
517 **
518 **
519 **
520 **
521 **
522 **
523 **
524 **
525 **
526 **
527 **
528 **
529 **
530 **
531 **
532 **
533 **
534 **
535 **
536 **
537 **
538 **
539 **
540 **
541 **
542 **
543 **
544 **
545 **
546 **
547 **
548 **
549 **
550 **
551 **
552 **
553 **
554 **
555 **
556 **
557 **
558 **
559 **
560 **
561 **
562 **
563 **
564 **
565 **
566 **
567 **
568 **
569 **
570 **
571 **
572 **
573 **
574 **
575 **
576 **
577 **
578 **
579 **
580 **
581 **
582 **
583 **
584 **
585 **
586 **
587 **
588 **
589 **
590 **
591 **
592 **
593 **
594 **
595 **
596 **
597 **
598 **
599 **
600 **
601 **
602 **
603 **
604 **
605 **
606 **
607 **
608 **
609 **
610 **
611 **
612 **
613 **
614 **
615 **
616 **
617 **
618 **
619 **
620 **
621 **
622 **
623 **
624 **
625 **
626 **
627 **
628 **
629 **
630 **
631 **
632 **
633 **
634 **
635 **
636 **
637 **
638 **
639 **
640 **
641 **
642 **
643 **
644 **
645 **
646 **
647 **
648 **
649 **
650 **
651 **
652 **
653 **
654 **
655 **
656 **
657 **
658 **
659 **
660 **
661 **
662 **
663 **
664 **
665 **
666 **
667 **
668 **
669 **
670 **
671 **
672 **
673 **
674 **
675 **
676 **
677 **
678 **
679 **
680 **
681 **
682 **
683 **
684 **
685 **
686 **
687 **
688 **
689 **
690 **
691 **
692 **
693 **
694 **
695 **
696 **
697 **
698 **
699 **
700 **
701 **
702 **
703 **
704 **
705 **
706 **
707 **
708 **
709 **
710 **
711 **
712 **
713 **
714 **
715 **
716 **
717 **
718 **
719 **
720 **
721 **
722 **
723 **
724 **
725 **
726 **
727 **
728 **
729 **
730 **
731 **
732 **
733 **
734 **
735 **
736 **
737 **
738 **
739 **
740 **
741 **
742 **
743 **
744 **
745 **
746 **
747 **
748 **
749 **
750 **
751 **
752 **
753 **
754 **
755 **
756 **
757 **
758 **
759 **
760 **
761 **
762 **
763 **
764 **
765 **
766 **
767 **
768 **
769 **
770 **
771 **
772 **
773 **
774 **
775 **
776 **
777 **
778 **
779 **
780 **
781 **
782 **
783 **
784 **
785 **
786 **
787 **
788 **
789 **
790 **
791 **
792 **
793 **
794 **
795 **
796 **
797 **
798 **
799 **
800 **
801 **
802 **
803 **
804 **
805 **
806 **
807 **
808 **
809 **
810 **
811 **
812 **
813 **
814 **
815 **
816 **
817 **
818 **
819 **
820 **
821 **
822 **
823 **
824 **
825 **
826 **
827 **
828 **
829 **
830 **
831 **
832 **
833 **
834 **
835 **
836 **
837 **
838 **
839 **
840 **
841 **
842 **
843 **
844 **
845 **
846 **
847 **
848 **
849 **
850 **
851 **
852 **
853 **
854 **
855 **
856 **
857 **
858 **
859 **
860 **
861 **
862 **
863 **
864 **
865 **
866 **
867 **
868 **
869 **
870 **
871 **
872 **
873 **
874 **
875 **
876 **
877 **
878 **
879 **
880 **
881 **
882 **
883 **
884 **
885 **
886 **
887 **
888 **
889 **
890 **
891 **
892 **
893 **
894 **
895 **
896 **
897 **
898 **
899 **
900 **
901 **
902 **
903 **
904 **
905 **
906 **
907 **
908 **
909 **
910 **
911 **
912 **
913 **
914 **
915 **
916 **
917 **
918 **
919 **
920 **
921 **
922 **
923 **
924 **
925 **
926 **
927 **
928 **
929 **
930 **
931 **
932 **
933 **
934 **
935 **
936 **
937 **
938 **
939 **
940 **
941 **
942 **
943 **
944 **
945 **
946 **
947 **
948 **
949 **
950 **
951 **
952 **
953 **
954 **
955 **
956 **
957 **
958 **
959 **
960 **
961 **
962 **
963 **
964 **
965 **
966 **
967 **
968 **
969 **
970 **
971 **
972 **
973 **
974 **
975 **
976 **
977 **
978 **
979 **
980 **
981 **
982 **
983 **
984 **
985 **
986 **
987 **
988 **
989 **
990 **
991 **
992 **
993 **
994 **
995 **
996 **
997 **
998 **
999 **
1000 **

```

09/01/76 PAGE 41

LINE CACI SIMSCRIPT II.5 RELEASE AF

```

155 LET POT.CRK = CCL + ((POT.CRK-CCL)/M1.MEAN) * M2.MEAN
156 REGARDLESS
157 IF MAX.CRK GT CCL
158 LET POT.CRK = MAX.CRK + 42.444 * ARCO(L10)
159 REGARDLESS
160 IF POT.STA.REO LT POT.CRK + STW.REO - MAX.CRK
161 LET POT.STA.REO = POT.CRK + STW.REO - MAX.CRK
162 REGARDLESS
163 REGARDLESS
164 REGARDLESS
165 LOOP
166 **
167 ** EFFECTS FOUND DURING THIS INSPECTION CAN CAUSE AN ADDITIONAL INCREASE IN THE
168 ** FREQUENCY OF NORMALLY SCHEDULED INSPECTIONS
169 **
170 IF POT.STA.REO GT FRF.LST/2.0
171 SCHEDULE AN INCREASE IN INSPECTION FREQUENCY FOR
172 REGARDLESS
173 **
174 RETURN
175 END

```

LOCAL VARIABLES OF THIS ROUTINE

AREA	DATA	TYPE	VALUE	ADDRESS
CCL	0.000	DOUBLE	0.000	0000 15
L1	0.000	DOUBLE	0.000	0000 25
L10	0.000	DOUBLE	0.000	0000 35
L11	0.000	DOUBLE	0.000	0000 45
L12	0.000	DOUBLE	0.000	0000 55
L13	0.000	DOUBLE	0.000	0000 65
L14	0.000	DOUBLE	0.000	0000 75
L15	0.000	DOUBLE	0.000	0000 85
L16	0.000	DOUBLE	0.000	0000 95
L17	0.000	DOUBLE	0.000	0001 05
L18	0.000	DOUBLE	0.000	0001 15
L19	0.000	DOUBLE	0.000	0001 25
L20	0.000	DOUBLE	0.000	0001 35
L21	0.000	DOUBLE	0.000	0001 45
L22	0.000	DOUBLE	0.000	0001 55
L23	0.000	DOUBLE	0.000	0001 65
L24	0.000	DOUBLE	0.000	0001 75
L25	0.000	DOUBLE	0.000	0001 85
L26	0.000	DOUBLE	0.000	0001 95
L27	0.000	DOUBLE	0.000	0002 05
L28	0.000	DOUBLE	0.000	0002 15
L29	0.000	DOUBLE	0.000	0002 25
L30	0.000	DOUBLE	0.000	0002 35
L31	0.000	DOUBLE	0.000	0002 45
L32	0.000	DOUBLE	0.000	0002 55
L33	0.000	DOUBLE	0.000	0002 65
L34	0.000	DOUBLE	0.000	0002 75
L35	0.000	DOUBLE	0.000	0002 85
L36	0.000	DOUBLE	0.000	0002 95
L37	0.000	DOUBLE	0.000	0003 05
L38	0.000	DOUBLE	0.000	0003 15
L39	0.000	DOUBLE	0.000	0003 25
L40	0.000	DOUBLE	0.000	0003 35
L41	0.000	DOUBLE	0.000	0003 45
L42	0.000	DOUBLE	0.000	0003 55
L43	0.000	DOUBLE	0.000	0003 65
L44	0.000	DOUBLE	0.000	0003 75
L45	0.000	DOUBLE	0.000	0003 85
L46	0.000	DOUBLE	0.000	0003 95
L47	0.000	DOUBLE	0.000	0004 05



09/01/74 PAGE 82

LINE CACI SCHEDULE II.5 RELEASE RF

```

1  EVENT DECISION, ON, MOD
2
3  **
4  ** DECISION IS MADE TO IMPLEMENT A STRUCTURAL MODIFICATION WHENEVER
5  ** MCPH(MPAIR COST PER HOUR) + ICPH(INSPECTION COST PER HOUR)
6  ** IS GREATER THAN MCPH(MODIFICATION COST PER HOUR)
7
8  **
9  **
10 **
11 **
12 **
13 **
14 **
15 **
16 **
17 **
18 **
19 **
20 **
21 **
22 **
23 **
24 **
25 **
26 **
27 **
28 **
29 **
30 **
31 **
32 **
33 **
34 **
35 **
36 **
37 **
38 **
39 **
40 **
41 **
42 **
43 **
44 **
45 **
46 **
47 **
48 **
49 **
50 **
51 **
52 **
53 **

```

1 \*\*  
 2 \*\*  
 3 \*\*  
 4 \*\*  
 5 \*\*  
 6 \*\*  
 7 \*\*  
 8 \*\*  
 9 \*\*  
 10 \*\*  
 11 \*\*  
 12 \*\*  
 13 \*\*  
 14 \*\*  
 15 \*\*  
 16 \*\*  
 17 \*\*  
 18 \*\*  
 19 \*\*  
 20 \*\*  
 21 \*\*  
 22 \*\*  
 23 \*\*  
 24 \*\*  
 25 \*\*  
 26 \*\*  
 27 \*\*  
 28 \*\*  
 29 \*\*  
 30 \*\*  
 31 \*\*  
 32 \*\*  
 33 \*\*  
 34 \*\*  
 35 \*\*  
 36 \*\*  
 37 \*\*  
 38 \*\*  
 39 \*\*  
 40 \*\*  
 41 \*\*  
 42 \*\*  
 43 \*\*  
 44 \*\*  
 45 \*\*  
 46 \*\*  
 47 \*\*  
 48 \*\*  
 49 \*\*  
 50 \*\*  
 51 \*\*  
 52 \*\*  
 53 \*\*

09/01/76 PAGE 43

LINE CACT STMSCHMT 11.5 RELEASE NF

```

54 ALWAYS
55 LET TWO_NOD_SCH = "YES"
56 LET OVERLAP_SCH = "YES"
57 IF TEST_FAILURE = "YES" AND TIME = 0 START TEST + ACTUAL_AVG_FAIL_LIFE /
58 TEST_ACCEPTANCE
59 LET TEST_FAILURE = "NO"
60 CANCEL THE IMPLEMENTATION
61 RESTART THE IMPLEMENTATION
62 RELEASE
63 RELEASE
64 END

```

LOCAL VARIABLES IF THIS ROUTINE

ACCUMULATE	DOUBLE	WORD 14	ICPM	DOUBLE	WORD 15
L-4	INTERM	WORD 7	ICPM	DOUBLE	WORD 17
SP-CUST	DOUBLE	WORD 5	ICPM	DOUBLE	WORD 9
NFTS	DOUBLE	WORD 21	ICPM	DOUBLE	WORD 1
POST-MOD-M	DOUBLE	WORD 11	ICPM	DOUBLE	WORD 19
TUNING	DOUBLE	WORD 3			

LINE CACI SCRIPT II.5 RELEASE OF 09/01/76 PAGE 84

```

1 EVENT IMPLEMENTATION MODIFICATION
2 **
3 ** REPRESENTS DEVELOPMENT OF MODIFICATION BECAUSE OF SERVICE EXPERIENCE
4 **
5 LET MOD = MOD + 1
6 LET MOD = 0
7 LET TIME_MODIFIED = 1000
8 LET REPAIR_MODIFIED = 0
9 LET IMPROV_SCH = "NO"
10 LET ARCD(LIL) = 1000(LIL)
11 LET ARCD(LIL) = 1000(LIL)
12
13 IF ARCD(LIL) > 0
14   JUMP ARCD
15 OTHERWISE
16   LET SCIN(MC) = ARCD(LIL)
17   LET SCIN(MC) = ARCD(LIL)
18   HERE
19   LET COST_OF_REPAIRS = 0.0
20   LET AIRFRAME_TIME = 0.0
21   LET MODIFIED = "NO"
22   LET MOD = MOD + ((1.0-MOD)*.15)
23   LET MOD = MOD + .15
24   CALL EVALUATE(COST_OF_REPAIRS,PREDICTED_LIFE) YIELDING ACTUAL_AVG_FAT_LIFE
25   JUMP ARCD
26 OTHERWISE
27   LET ACTUAL_AVG_FAT_LIFE = PREDICTED_LIFE
28   HERE
29   FOR EVERY AIRCRAFT IN ACTIVE_FLEET
30     LET SCIN(PREDICTED_LIFE_ID) = "YES"
31   NEXT
32   HERE
33   HERE
34   END

```

LOCAL VARIABLES OF THIS ROUTINE

NAME	TYPE	VALUE	SCOPE
MOD	INTEGER	0	LOCAL
TIME_MODIFIED	INTEGER	1000	LOCAL
REPAIR_MODIFIED	INTEGER	0	LOCAL
IMPROV_SCH	STRING	"NO"	LOCAL
ARCD	ARRAY		LOCAL
COST_OF_REPAIRS	REAL	0.0	LOCAL
AIRFRAME_TIME	REAL	0.0	LOCAL
MODIFIED	STRING	"NO"	LOCAL
ACTUAL_AVG_FAT_LIFE	REAL		LOCAL
PREDICTED_LIFE	REAL		LOCAL
ACTIVE_FLEET	ARRAY		LOCAL
SCIN	ARRAY		LOCAL
SCIN_ID	ARRAY		LOCAL

- 204 -

- 205 -

— — — — —  
— — — — —

STILLING SMITH & SONS LTD

- 206 -

LINE CACI SCRIPT 11.5 RELEASE AF

```

COUNTING SUMMARY
  OFFIVE XRGF AS AN INTEGER VARIABLE
  START NEW PAGE
  WRITE MODEL(1), MODEL(2) AS S 50, "AIRCRAFT TYPE: ", ? A Q
  SKIP 2 OUTPUT LINES
  PRINT 2 DOUBLE LINE WITH SIZE OF FLEET, USAGE, LIFE AS FOLLOWS
  NUMBER OF AIRCRAFT IN FLEET: ***** AIRCRAFT SERVICE LIFE: ***** MONTHS

```

AIRCRAFT SERVICE LIFE: \*\*\*\*\* HOURS

27. 28. 29.

POINT 1 DUBAIFITNF AS FILLING

NUMBER AND TIME TO INITIATION OF AIRCRAFT DEFECTS

```

SKIP 1 OUTPUT LINE
IF GOSUB GT 0      LET PSICO = SICR      IFT PLICR = LICR      ALWAYS
IF GOSUB GT 0      LET PSICO = SICR      IFT PLICO = LICR      ALWAYS
IF GOSUB GT 0      LET PSICO = SICR      IFT PLISD = LISD      ALWAYS
PRINT "ONLY LE LINES WITH GOSUB, GOSUB, GOSUB, GOSUB, GOSUB, PSICO, PSICO, PSICO, PSICO, PLISD, PLISD, AICR, AICR, ALSO AS FOLLOWS

```

	FIRST CRACK	COMPLETION	SERVICE DURATION	PERMANENT DEFECTS
FREQUENCIES	*****	*****	*****	*****
MIN.(HRS)	*****	*****	*****	*****
MAX(HRS)	*****	*****	*****	*****
AVG(HRS)	*****	*****	*****	*****

SLIP 2 INPUT LINES  
PATENT 1 DOUBLE LINE  
AS FOLLOWS  
NUMBER AND LENGTH OF CRACKS DETECTED AT EACH LEVEL OF INSPECTION.

[illegible]

OCURRENCES	A-LEVEL	M-LEVEL	C-LEVEL	D-LEVEL	SPT. LEVEL
MIN(12)	*****	*****	*****	*****	*****
MAX(12)	*****	*****	*****	*****	*****
MAX(10)	*****	*****	*****	*****	*****
AUG(12)	*****	*****	*****	*****	*****
SUM(12)	*****	*****	*****	*****	*****

```

      (1) SKIP 2 OUTPUT LINES
      PRINT 1 DOUBLE LINE AS FOLLOWS
      JUMPER AND DATA (1) COMMISSION DEFECTS DETECTED AT FACT LEVEL IN PRODUCTION
      *****
      ****
      ****
      ****

```

[illegible]

2

- 208 -



THIS PAGE IS BEST QUALITY PRACTICABLE  
FROM COPY FURNISHED TO DDC

00/01/70 PAGE 90

LINE CACT STMSCHPT 11.5 RELEASE RF

LOCAL VARIABLES OF THIS ROUTINE

I-1	INTEGER	WORD 5	I-2	INTEGER	WORD 6
I-3	INTEGER	WORD 7	J-1	INTEGER	WORD 8
K-1	INTEGER	WORD 10	K-2	INTEGER	WORD 11
L-1	INTEGER	WORD 12	K-4	INTEGER	WORD 13
L-11	INTEGER	WORD 14	L-11	INTEGER	WORD 32
L-13	INTEGER	WORD 53	L-15	INTEGER	WORD 54
L-17	INTEGER	WORD 75	L-19	INTEGER	WORD 76
L-21	INTEGER	WORD 77	L-3	INTEGER	WORD 15
L-5	INTEGER	WORD 14	L-7	INTEGER	WORD 17
L-9	INTEGER	WORD 31	K-11	INTEGER	WORD 9
PLACA	DOUBLE	WORD 57	PLACW	DOUBLE	WORD 35
PLBCA	DOUBLE	WORD 61	PLCCW	DOUBLE	WORD 39
PLCCA	DOUBLE	WORD 65	PLCCW	DOUBLE	WORD 43
PLCCA	DOUBLE	WORD 69	PLCCW	DOUBLE	WORD 47
PLSCA	DOUBLE	WORD 73	PLSCW	DOUBLE	WORD 51
PLISD	DOUBLE	WORD 25	PLISW	DOUBLE	WORD 21
PLISD	DOUBLE	WORD 29	PSACA	DOUBLE	WORD 55
PSACR	DOUBLE	WORD 33	PSACA	DOUBLE	WORD 59
PSACR	DOUBLE	WORD 37	PSCCA	DOUBLE	WORD 63
PSCCR	DOUBLE	WORD 41	PSCCA	DOUBLE	WORD 67
PSCCR	DOUBLE	WORD 45	PSCCA	DOUBLE	WORD 71
PSCCR	DOUBLE	WORD 49	PSYCH	DOUBLE	WORD 23
PSICH	DOUBLE	WORD 19	PSYCH	DOUBLE	WORD 27
K-1	DOUBLE	WORD 3	WGE	INTEGER	WORD 1